



Universitat Politècnica de Catalunya·BarcelonaTech

Grau en Enginyeria en Informàtica - Tecnologies de la informació

Treball de Final de Grau

---

# Disseny i realització d'un sistema de control i monitoratge remot per a una xarxa distribuïda de sensors mediambientals

---

Aina Soler Gàlvez

Director: Jose Maria Barceló Ordinas

Codirector: Jorge Garcia Vidal

Q2 2016-2017



# Agraïments

Agrair a tots els companys del laboratori l'ajuda prestada durant aquests mesos, així com al director del projecte, Jose Maria Barceló, el codirector, Jorge Garcia, i a tots els altres membres del projecte CAPTOR per l'ajuda prestada i la oportunitat de participar en el projecte. Gràcies també a les porteres del CSIC, per aguantar tots els viatjats que hi he fet durant aquests mesos.

## **Resum**

L'objectiu d'aquest projecte son 3 tasques diferenciades: Detectar, analitzar i solucionar problemes en els nodes del projecte Europeu CAPTOR; Dissenyar un sistema d'eines, i una interfície senzilla d'usar per l'equip que es dedicarà a mantenir els nodes; I finalment, com a extensió del punt anterior, dissenyar una solució tecnològica per tal de que el mètode d'accés a Internet dels nodes sigui transparent, simplificant-ne el seu manteniment.

## **Resumen**

El objetivo de este proyecto son 3 tareas diferenciadas: Detectar, analizar y solucionar problemas en los nodos del proyecto Europeo CAPTOR; Diseñar un sistema de herramientas, y una interfaz sencilla de usar para el equipo que se dedicará a mantener los nodos; Y finalmente, como extensión del punto anterior, diseñar una solución tecnológica para que el método de acceso a Internet de los nodos sea transparente, simplificando su mantenimiento.

## **Abstract**

The goal of this project are three distinct tasks: Identify, analyze and solve problems in the CAPTOR nodes, from the European project CAPTOR; Design a system of tools and a simple interface for the team that will be maintaining the nodes; And finally, as an extension of the previous point, to design a technological solution so the Internet access of the nodes is transparent to the maintainers, simplifying its maintenance.



# Índex

<b>1</b>	<b>Introducció, contingut i estructura de la memòria</b>	<b>1</b>
1.1	Context en el que es desenvolupa el treball . . . . .	1
1.2	Problemes a resoldre . . . . .	2
1.3	Solucions proposades i implementades . . . . .	2
1.4	Estructura de la memòria . . . . .	3
1.4.1	Introducció, Gestió de projectes i Sostenibilitat . . . . .	3
1.4.2	Anàlisi i millora del node CAPTOR . . . . .	3
1.4.3	Interfície web de gestió . . . . .	3
1.4.4	Accés remot a nodes fora de la xarxa publica . . . . .	4
<b>2</b>	<b>Introducció</b>	<b>5</b>
2.1	Formulació del problema . . . . .	5
2.2	Context . . . . .	8
2.2.1	Parts interessades . . . . .	9
2.2.2	Estat Actual . . . . .	9
<b>3</b>	<b>Gestió de projectes</b>	<b>11</b>
3.1	Estat de l'art . . . . .	11
3.1.1	Servidor d'ordres . . . . .	12
3.1.2	CAPTOR-status diari . . . . .	13
3.1.3	Emnify . . . . .	13
3.2	Abast del projecte . . . . .	14
3.3	Metodologia i rigor . . . . .	14
3.4	Descripció de les tasques . . . . .	15

3.4.1	Duració del projecte . . . . .	15
3.4.2	Tasques . . . . .	15
3.4.3	Dependències . . . . .	17
3.4.4	Recursos . . . . .	17
3.5	Valoració d'alternatives i pla d'acció . . . . .	19
3.5.1	Riscos . . . . .	19
3.5.2	Alternatives i pla d'acció . . . . .	20
3.6	Identificació y estimació dels costos . . . . .	20
3.6.1	Recursos humans . . . . .	20
3.6.2	Hardware . . . . .	21
3.6.3	Software i llicències . . . . .	21
3.6.4	Altres costos . . . . .	22
3.6.5	Resum de costos . . . . .	22
3.7	Control de gestió . . . . .	22
3.8	Desviacions de la planificació . . . . .	23
3.9	Modificacions sobre el plantejament inicial . . . . .	23
<b>4</b>	<b>Sostenibilitat</b>	<b>25</b>
4.1	Sostenibilitat i compromís social . . . . .	25
4.1.1	Impacte ambiental . . . . .	25
4.1.2	Impacte econòmic . . . . .	26
4.1.3	Impacte Social . . . . .	26
<b>5</b>	<b>Anàlisi i millora del node CAPTOR</b>	<b>27</b>
5.1	Introducció . . . . .	27
5.2	Objectius . . . . .	31
5.3	Tecnologies . . . . .	31
5.3.1	Arduino . . . . .	31
5.3.2	OpenWRT . . . . .	33
5.3.3	EAGLE . . . . .	33
5.3.4	Fabricació manual de PCB . . . . .	33

---

5.4	Desenvolupament . . . . .	35
5.4.1	Proposta del problema . . . . .	35
5.4.2	Disseny de la solució . . . . .	36
5.4.3	Fabricació del prototip . . . . .	38
5.4.4	Test del prototip . . . . .	39
5.4.5	Disseny i fabricació de la placa final . . . . .	40
5.5	Anàlisi de resultats . . . . .	42
<b>6</b>	<b>Interfície web de gestió</b>	<b>45</b>
6.1	introducció . . . . .	45
6.2	Objectius . . . . .	45
6.3	Tecnologies . . . . .	46
6.3.1	Python . . . . .	46
6.3.2	Flask . . . . .	46
6.3.3	HTML . . . . .	47
6.3.4	Javascript . . . . .	48
6.3.5	Server-Sent Events . . . . .	49
6.3.6	NodeJS, WebSockets i Wetty . . . . .	50
6.3.7	NGINX i uWSGI . . . . .	51
6.3.8	Let's Encrypt i Certificats digitals . . . . .	52
6.3.9	PostgreSQL . . . . .	52
6.3.10	SQLAlchemy . . . . .	53
6.4	Desenvolupament . . . . .	53
6.4.1	Disseny de l'estructura de la pàgina . . . . .	53
6.4.2	Disseny de la base de dades . . . . .	55
6.4.3	Posada a punt de l'entorn . . . . .	58
6.4.4	Flask: Servint les pàgines . . . . .	61
6.4.5	Usuaris i Administradors . . . . .	64
6.4.6	Afegir els CAPTORs . . . . .	68
6.4.6.1	Formulari d'alta . . . . .	68
6.4.6.2	API de Google maps . . . . .	70

6.4.6.3	Consultar les dades del CAPTOR . . . . .	72
6.4.7	Editar i eliminar CAPTORs . . . . .	73
6.4.8	Dashboard . . . . .	74
6.4.8.1	Status del CAPTOR al sistema . . . . .	74
6.4.8.2	API d'Emnify: Status del CAPTOR a la xarxa cel·lular . . . . .	79
6.4.9	Connexió directa per SSH a través de la web . . . . .	80
6.4.10	Deployment i HTTPS . . . . .	82
6.4.10.1	Servint l'aplicació amb uWSGI . . . . .	82
6.4.10.2	Iniciar l'aplicació des de systemd . . . . .	83
6.4.10.3	Configurant NGINX per passar les peticions . . . . .	84
6.4.10.4	Obtenint els certificats . . . . .	85
6.4.10.5	Generant un Diffie-Hellman Group segur . . . . .	86
6.4.10.6	Assegurant la nostra pàgina web amb TLS/SSL . . . . .	86
6.5	Anàlisi de resultats . . . . .	89
<b>7</b>	<b>Accés remot a nodes fora de la xarxa pública</b>	<b>91</b>
7.1	introducció . . . . .	91
7.2	Objectius . . . . .	92
7.3	Desenvolupament . . . . .	92
7.3.1	Reverse SSH Tunneling . . . . .	92
7.3.2	Configuració d'SSH . . . . .	94
7.3.3	Automatització de la connexió del CAPTOR . . . . .	95
7.3.3.1	Monitoritzant la connexió amb AutoSSH . . . . .	96
7.3.3.2	Daemon per iniciar la connexió . . . . .	96
7.3.3.3	Claus RSA i Known Hosts . . . . .	99
7.3.3.4	Problemes amb la connexió . . . . .	99
7.4	Alternativa: SIM Emnify . . . . .	100
7.5	Anàlisi de resultats . . . . .	101
<b>8</b>	<b>Conclusions</b>	<b>103</b>
8.1	Conclusions . . . . .	103

8.1.1	Relació amb el grau . . . . .	103
8.1.2	Aspectes a millorar . . . . .	104
8.2	Opinió Personal . . . . .	105
8.3	Propostes de Terball Futur . . . . .	105
<b>Bibliografia</b>		<b>106</b>
<b>Índex de figures</b>		<b>109</b>
<b>Índex de codi</b>		<b>113</b>



# Capítol 1

## Introducció, contingut i estructura de la memòria

### 1.1 Context en el que es desenvolupa el treball

El 2016 es va iniciar el projecte CAPTOR, un projecte que combina ciència ciutadana, institucions científiques i activistes socials i mediambientals per augmentar el coneixement i trobar solucions a problemes relacionats amb la contaminació de l'aire. El projecte CAPTOR utilitza sensors de molt baix cost per recopilar dades a 3 regions Europees altament afectades per la contaminació d'ozó: Catalunya, Itàlia i Àustria. Aquests sensors son mantinguts per ciutadans i voluntaris. Les campanyes de preses de dades es realitzen al estiu, havent sigut la primera al 2016, la segona el 2017 i la tercera i final es durà a terme l'estiu del 2018

La UPC, en col·laboració amb el CSIC, Ecologistas en Acción i la fundació Guifi.Net, son els col·laboradors espanyols en el projecte.

Al Setembre de 2016 m'uneixo amb una beca de col·laboració al Statistical Analysis of Networks and Systems (SANS) Research Group del DAC, a la UPC, que treballa dissenyant i muntant aquests sensors, per tal d'ajudar a trobar solucions i realitzar nous sistemes per superar problemes detectats en els nodes durant la campanya del 2016. Aquest treball recull les solucions proposades i implementades com a projecte de final de grau dins d'aquest grup.

## 1.2 Problemes a resoldre

Els 3 principals problemes que s'han volgut resoldre son els punts següents:

- **Inestabilitat en la presa de dades amb el moviment dels sensors:** Es detecta que al moure els nodes, ja sigui al transportat-los o per la climatologia, les dades preses canvien de rang dinàmic.
- **Impossibilitat de manipular el software dels sensors a distància:** Un cop col·locats els sensors a la seva localització final, no es té forma pràctica de manipular i administrar el seu sistema operatiu des del laboratori de Barcelona.
- **Inexistència d'una eina específica per facilitar les tasques de manteniment:** Els administradors dels nodes han d'usar una gran varietat d'eines i documents per portar un registre i administrar cada un dels nodes.

## 1.3 Solucions proposades i implementades

A continuació, un resum de les 3 solucions que s'han implementat per resoldre cada un dels problemes anteriors. La seva implementació i funcionament en detall s'especifica a cada un dels capítols corresponents.

- **Inestabilitat en la presa de dades amb el moviment dels sensors:** Es realitza un disseny i placa integrada amb els components electrònics soldats. Es fan tests amb prototips fabricats manualment i es fabriquen finalment de forma professional les plaques a les que es solden els components manualment. En detall en el Capítol 5.
- **Impossibilitat de manipular el software dels sensors a distància:** Es dissenya un sistema de túnels encapsulats en connexions SSH passives, que creen canals de comunicació virtuals entre els clients i els nodes a través d'un servidor intermediari, sense augmentar el consum de dades per part del node si la connexió no està en us. En detall en el Capítol 7.
- **Inexistència d'una eina específica per facilitar les tasques de manteniment:** Es dissenya i implementa una pàgina web que inclou en una sola les diverses eines necessàries per l'administració i monitorització dels nodes. Des d'un sistema d'informació de contacte del voluntari, fins a un mètode per establir una connexió SSH amb els nodes a través del navegador sense necessitat de disposar de client SSH en el dispositiu usat, passant per un panell de control amb tota la informació d'un cop d'ull i mapes per veure la localització dels nodes. En detall en el Capítol 6.



## 1.4 Estructura de la memòria

Ja que el treball té 4 parts molt diferenciades, hem decidit separar cada una d'elles en capítols independents, per tal de fer que la informació es trobi més organitzada.

La primera part correspon al contingut de l'assignatura de gestió de projectes, de les 3 restants cada part tindrà de forma independent, informació sobre els objectius, tecnologies, forma d'implementació i anàlisi de resultats. Finalment, en el últim capítol de conclusions, es veuran junts els resultats de cada un dels capítols anteriors i es trauran conclusions d'ells.

Específicament, el contingut de cada un dels capítols es el següent:

### 1.4.1 Introducció, Gestió de projectes i Sostenibilitat

Aquests capítols corresponen a l'assignatura de Gestió de Projectes. En el capítol d'Introducció, veurem el context en el que desenvolupem el treball, i es presenta el material i els nodes amb els que treballarem. En el capítol de gestió de projectes, es fa un anàlisi de temps i costos econòmics del desenvolupament del treball. Finalment en el capítol de Sostenibilitat, es fa un anàlisi de l'impacte social i ambiental del treball. Aquest contingut és l'apartat preliminar realitzat abans del desenvolupament del treball, i alguns apartats han canviat durant el desenvolupament, aquest canvis es descriuen en els últims apartats del capítol de Gestió de Projectes.

### 1.4.2 Anàlisi i millora del node CAPTOR

En aquest capítol, veurem quins problemes s'han detectat als nodes CAPTOR de la campanya de l'any 2016, quines solucions es van proposar, com es van provar i implementar, i finalment veurem els resultats obtinguts duran aquest inici de campanya 2017, analitzant si les solucions implementades han ajudat a millorar els resultats.

### 1.4.3 Interfície web de gestió

En el capítol d'Interfície web de gestió, veurem com s'ha implementat la interfície web per ajudar a gestionar els nodes, des del disseny de la seva base de dades fins a la implementació i test final. Així com l'estructura del servidor que l'allotja i les tecnologies que la suporten.

#### 1.4.4 Accés remot a nodes fora de la xarxa publica

Finalment, es veurà la solució implementada per poder accedir als nodes que es troben en les xarxes privades dels voluntaris o que bé estan funcionant a través de la xarxa de telefonia mòbil. És a dir, com ens podem comunicar de forma ràpida i eficient amb aquells nodes que no disposen d'adreça publica. També veurem una comparativa amb l'altre opció existent.

# Capítol 2

## Introducció

### 2.1 Formulació del problema

En les últimes dècades, la contaminació de l'aire en el nostre entorn ha augmentat considerablement. Aquesta contaminació atmosfèrica, la presència en l'aire de matèria que pot ser perjudicial per a la salut dels éssers vius, ha arribat a un nivell important i preocupant amb efectes notables en la salut dels ciutadans [1].

Existeixen diferents organismes que s'encarreguen de regular aquesta contaminació, sent la Unió Europea i la OMS els dos organismes amb més influència en el nostre país. Aquestes organitzacions disposen d'una sèrie d'estacions de referència (Com la de la figura 2.1), que són instal·lades en punts notables de les ciutats i s'encarreguen de mesurar diferents paràmetres de l'aire, especialment ozó (O<sub>3</sub>), òxids de nitrogen, partícules sòlides, monòxid de carboni i òxids de sofre [2].



Figura 2.1: Estació de referència

**Font:** Departament de Territori i Sostenibilitat

D'aquests elements, el ozó ( $O_3$ ) es especialment perjudicial quan es troba a altituds mes baixes que la de la coneguda com a capa d'ozó, necessària per protegir els éssers vius de la radiació ultraviolada. A aquest ozó l'anomenem ozó troposfèric i provoca notables problemes respiratoris a la població [3].

L'ozó troposfèric es un contaminant secundari, ja que no es emes directament si no que es creat per una reacció química en la que l'oxigen ( $O_2$ ), els òxids de nitrogen ( $NO_x$ ) i alguns compostos orgànics en suspensió son afectats per la llum solar i formen  $O_3$  (Representat a la figura 2.2). Per tant, durant l'estiu, al ser l'època de l'any amb mes incidència solar, els nivells d'ozó ascendeixen de forma important. Com ja hem mencionat, l' $O_3$  es forma a partir de  $O_2$ ,  $NO_x$  i compostos orgànics, aquests elements, productes de l'activitat industrial als nuclis urbans, s'escampen pel territori, arribant a zones rurals on l'ozó es especialment perjudicial al afectar sobre terreny agrícola i ramader [4].

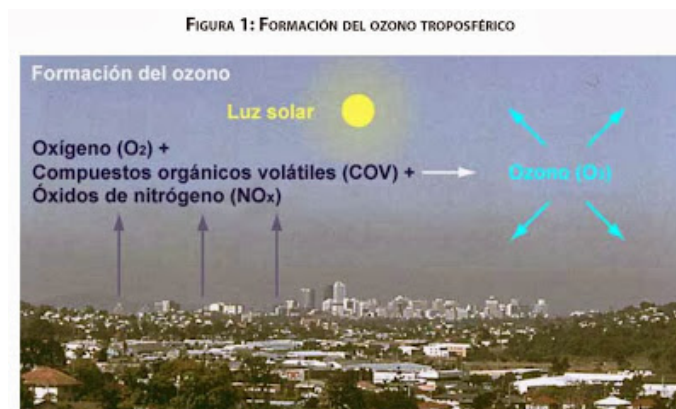


Figura 2.2: Formació de l'ozó troposfèric

**Font:** Ministerio de Agricultura, Alimentación y Medio Ambiente

A mes, l' $O_3$  tendeix a descompondre's en zones amb molta concentració d' $NO$ , un altre contaminant present en les ciutats, on es concentren la major part de les estacions de referència, cosa que fa que en les localitzacions on realment afecta el  $O_3$  provinent dels nuclis urbans, les zones rurals, no es disposi d'un seguiment exhaustiu d'aquest contaminant en les zones mes afectades. A la figura 2.3 es pot observar que la contaminació es concentra a les terres de Vic tot i generar-se a la costa, que no es veu afectada.

Degut a això, el Gener del 2016, s'inicia el projecte CAPTOR [5] finançat per la Comissió Europea dins el programa Horitzó 2020, en el que participa la UPC, el CSIC i diverses associacions ecologistes. Aquest projecte busca distribuir arreu del territori en localitzacions aportades per voluntaris ciutadans, una xarxa de petits sensors de baix cost (semblants

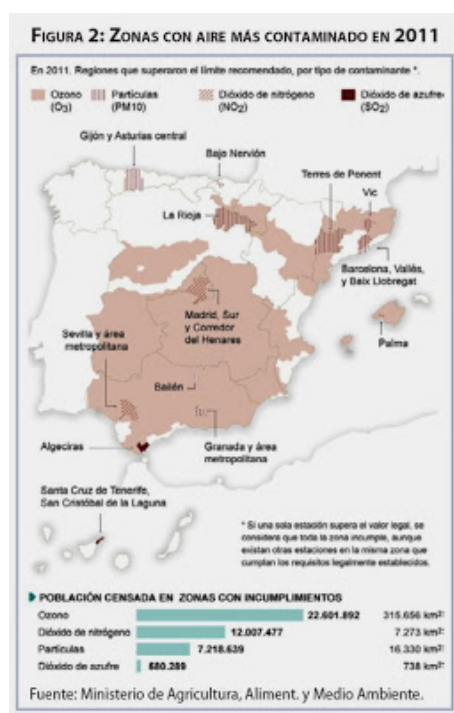


Figura 2.3: Zones amb l'aire més contaminat(2011)

**Font:** Ministerio de Agricultura, Alimentación y Medio Ambiente

al de la figura 2.4) que permetin recollir dades sobre l'ozó en les localitzacions més afectades.

Actualment el projecte CAPTOR disposa d'uns 40 sensors (amb previsió d'uns 30 addicionals per a l'estiu del 2017) repartits entre el centre de Catalunya i el nord D'Itàlia i Àustria. Aquests sensors publiquen de forma periòdica les dades, però realitzar modificacions i fer seguiment del seu funcionament a un temps curt es difícil ja que no es disposa de les eines per facilitar-ho, ni en els sensors ni en els dispositius dels usuaris.

Partint d'aquests problemes, l'objectiu d'aquest treball es centrarà en realitzar les eines i infraestructures necessàries per permetre tant un monitoratge remot i a temps real d'aquests sensors amb una interfície fàcil d'usar com el seu control i modificació a distància (Evitant els costos econòmics i impacte ecològic dels desplaçaments fins a la localització dels sensors). Així com s'intentaran sobrepassar els reptes tecnològics que deriven del fet de que la majoria de sensors es trobin en xarxes privades de voluntaris que no han de ser modificades, ja que no sempre els voluntaris seran persones expertes o amb els coneixements necessaris. Finalment, s'analitzaran problemes detectats en els sensors en campanyes anteriors, i es proposaran i duran a terme solucions.



Figura 2.4: Sensor CAPTOR

**Font:** Diari El Ter

## 2.2 Context

Aquest projecte sorgeix de la necessitat d'un sistema per monitoritzar i controlar a temps real els sensors de la xarxa de sensors CAPTOR. Es realitza dins del grup d'investigació SANS (Statistical Analysis of Networks and Systems) del departament d'Arquitectura de Comutadors, a la Universitat Politècnica de Barcelona.

La realització d'un Software de control i monitoratge gràfic i senzill per ser usat pels responsables de la xarxa de sensors que permeti veure l'estat en temps real de cada un dels sensors de la xarxa així com controlar-los de la forma més immediata possible es el principal objectiu del projecte. Això també comporta la realització d'una solució que integri els CAPTORs al programa de monitoratge, i el disseny i realització de tota la infraestructura "back-end" que haurà de permetre el seu correcte funcionament.

A part, també s'analitzaran els resultats de la campanya anterior, per tal de detectar els problemes que van sorgir i proposar i implementar solucions.

Es un projecte que integra tant Hardware (La programació del Hardware del CAPTOR, i la realització de les millores en el CAPTOR) com Software (El Software de control i el del servidor del programari de control), així com el disseny i integració de sistemes de xarxa (Les funcionalitats del Servidor i els protocols de comunicació que permetin sobrepassar els problemes generats per la localització dels sensors en cases de voluntaris)

### 2.2.1 Parts interessades

A continuació es descriuen les parts interessades en la realització del projecte, que en ordre de prioritats, són:

- **Responsables del projecte CAPTOR:** La possibilitat de controlar els sensors de forma remota permetrà estalviar innumerables hores i recursos en desplaçaments per realitzar tasques de manteniment i canvis en el sistema del CAPTOR. També permetrà detectar problemes en els sensors de forma prematura, evitant problemes més greus com serien la corrupció de dades, la pressa de dades errònies o la pèrdua de comunicació total amb el sensor.
- **La comunitat científica participant en el projecte CAPTOR:** La possibilitat de veure l'estat dels sensors en temps real mitjançant una interfície senzilla pensada per a no-experts serà d'utilitat pels científics que participen en l'anàlisi i interpretació de les dades preses pels sensors del projecte.
- **Associacions Ecologistes, voluntaris i població general:** Disposar d'una forma de veure l'estat dels sensors en temps real és una forma molt interessant per conscienciar i implicar a la població en el projecte.

### 2.2.2 Estat Actual

Actualment tots els sensors es troben o bé a l'estació de referència ubicada als jardins del CSIC, a Palau Reial o al laboratori del grup d'investigació, amb la excepció de 6 sensors que es troben a Itàlia i Àustria. Aquests sensors (i els que s'afegiran a ells) seran distribuïts a la primavera a les cases dels voluntaris, de cara a la preparació de la campanya de pressa de dades de l'estiu 2017. S'espera que tots els sensors disposin d'integració amb el nou sistema de control de cara al estiu.

Aquests sensors en l'actualitat no disposen de cap forma de control directa que permeti realitzar-hi tasques complexes, i moltes vegades es necessita un desplaçament físic fins a la localització del sensor per tal de realitzar les modificacions i manteniment.

Sí es disposa d'un sistema molt bàsic de control que permet realitzar-hi tasques molt simples i un sistema que de forma indirecta permet saber un estat molt general del CAPTOR, aquests sistemes seran descrits a l'apartat d'Estat de l'Art, del capítol de Gestió de projectes.

En el moment de la finalització del projecte, tots els sensors de la xarxa es trobaran repartits arreu del territori en les zones de presa de dades, així com en els diferents països de l'Unió Europea que participen en el projecte.



# Capítol 3

## Gestió de projectes

### 3.1 Estat de l'art

Actualment existeixen multitud de solucions propietàries de monitoratge de sistemes informàtics (per exemple PandoraFMS a la figura 3.1) però la gran totalitat estan orientats al monitoratge d'ordinadors i servidors convencionals. No existeix cap solució que s'adapti al nostre context: Nodes poc potents, que operen dins de xarxes informàtiques de les que no tenim control. Degut això es pren la decisió d'implementar un sistema a mida, que, potser tot i no ser tant complet com els sistemes comercials, s'adaptarà millor a les nostres necessitats.

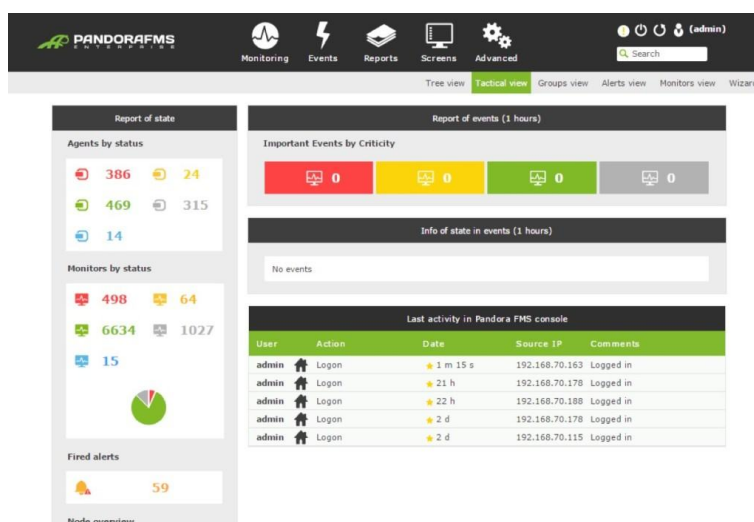


Figura 3.1: Sistema PandoraFMS de monitoratge

**Font:** PandoraFMS dev. blog

No obstant així, la majoria de sistemes de monitoratge, disposen d'unes interfícies i funcionalitats comunes [6], i ens podem basar en el disseny de sistemes d'èxit per a la realització del nostre sistema.

Com hem dit a l'apartat de context, actualment els CAPTORS disposen de dos sistemes molt limitats per tal de controlar-los i saber el seu estat, a continuació els descriurem i analitzarem perquè no son efectius i necessiten ser substituïts, així com què podem aprendre del temps que han estat en funcionament per al disseny del nou sistema de control i monitoratge. Finalment, parlarem del 3r sistema que ha aparegut durant el desenvolupament del projecte, i que en el transcurs del document, compararem amb la solució proposada.

### 3.1.1 Servidor d'ordres

Es disposa d'un servidor on els responsables poden allotjar ordres molt senzilles per al CAPTOR ("Executa aquest script...", "Envia aquestes dades altre cop..."). Un cop al dia, els sensors es connecten a aquest servidor i comproven si disposen d'ordres noves.

Es evident que aquest sistema no es l'ideal. Els CAPTORs nomes poden interpretar d'aquesta forma un nombre molt limitat d'ordres, i no es disposa de resposta immediata per part del sensor. En el pitjor dels casos, el sensor pot trigar fins a 24 hores en realitzar les ordres desitjades.

De les limitacions d'aquest mètode extraïem que el nou sistema:

- Ha de permetre realitzar tot tipus d'operacions en el CAPTOR, de la mateixa forma que es farien controlant-lo de forma no remota.
- He de rebre el feedback que genera el CAPTOR de les operacions que es realitzen sobre seu.
- Ha de permetre realitzar operacions sobre el CAPTOR sigui on sigui i sigui en el moment que sigui.
- Tot això s'ha de realitzar evitant la necessitat de que el CAPTOR faci espera activa, ocupant recursos i xarxa.

### 3.1.2 CAPTOR-status diari

Pel que fa al estatus de connexió del sensor, actualment es disposa d'un sistema molt bàsic. Simplement, un script, al final de cada dia, comprova quins sensors han enviat dades al servidor durant el dia, i genera un informe indicant si el sensor ha funcionat durant el dia actual o no ho ha fet.

Aquest sistema té els seus problemes: Si el sensor ha deixat de funcionar però ha enviat alguna dada abans de fer-ho, que és el cas més comú, es trigarà un dia sencer en detectar la fallada. A més, la única informació que s'obté d'aquest sistema és que durant un dia en concret el sensor no ha enviat cap dada al servidor.

De les limitacions d'aquest mètode extraïem que el nou sistema:

- Ha de tenir una granularitat més petita que "dia" fins al punt de saber en quines hores (o inclús minuts) el sensor no ha funcionat de la forma esperada.
- La informació ha de poder ser consultada de forma actualitzada en qualsevol moment.

### 3.1.3 Emnify

Com ja he mencionat, durant el transcurs del projecte, va aparèixer una 3a opció. Es va decidir canviar de proveïdor d'internet a Emnify [7]. Es tracta d'una operadora orientada a IoT, que ens permet usar la mateixa targeta amb la mateixa tarifa a qualsevol operador arreu del món.

La gran peculiaritat, però, és que totes les targetes contractades a internet d'aquesta empresa, a més a més de la seva adreça dins de la xarxa cel·lular tradicional, estan funcionant a través d'una VPN, a la que tenim accés.

Això ens ofereix avantatges respecte no tenir res, però creiem que hem pogut millorar aquests beneficis amb la nostra solució. A continuació llistem alguns problemes que encara existeixen amb aquesta solució i que solucionem en el nostre projecte:

- Alta complexitat. La connexió a la VPN és una configuració difícil i el seu ús i control dels nodes a través seu no és àpte per a persones amb pocs coneixements tecnològics.
- Poca portabilitat. Ja que la nostra solució està pensada per ser accessible en qualsevol dispositiu amb connexió a Internet i un navegador, i la configuració a la VPN s'ha de realitzar per a cadascun dels dispositius.

- La VPN afegeix un overhead considerable en el temps de comunicació i en les dades consumides.

Tot i així, la oportunitat d'utilitzar aquesta VPN durant el desenvolupament del projecte es molt positiva, ja que ens permet realitzar la majoria de tasques des del laboratori, estalviant temps i agilitzant gestions.

## 3.2 Abast del projecte

L'objectiu final es obtenir un sistema bàsic que permeti controlar i monitorejar de forma senzilla i intuïtiva els CAPTORs. No es pretén realitzar un gran Software amb multitud de funcionalitats, ja que no es disposa del tems ni dels recursos, sinó un programa útil i ben estructurat que realitzi totes les funcions bàsiques i sigui fàcil d'ampliar en un futur.

El principal obstacle al que s'enfronta el projecte, a part de la gran quantitat d'actors necessaris per al seu funcionament que s'han de realitzar, és les limitacions tecnològiques i de seguretat que ofereix el fet de que els CAPTORs estiguin funcionant en xarxes privades de voluntaris a les que no podem modificar les polítiques de seguretat. S'haurà de dissenyar una solució que permeti sobrepassar aquest fet, però implicarà sacrificar altres punts positius com la simplicitat del projecte, o la càrrega de treball en el servidor.

## 3.3 Metodologia i rigor

Per a la realització del projecte, s'utilitzarà un mètode iteratiu. Realitzant reunions periòdiques (Setmanals) amb els responsables del projecte CAPTOR (Que seran els usuaris finals del projecte) i el director del projecte, per tal d'anar analitzant les diferents funcionalitats i decidir els següents passos a seguir i els canvis necessaris.

S'utilitzaran a mes, unes eines de gestió de projectes durant el transcurs del projecte per tal de facilitar el seguiment per part del director i simplificar l'organització del codi i la documentació:

- **BitBucket:** Un repositori de codi que permet realitzar un control de versions i mantenir còpies de seguretat de tot el projecte, així com compartir-lo als futurs usuaris.
- **ShareLaTeX:** Un Software de col·laboració online en el que es redactarà la documentació del projecte, permetent així un seguiment mes senzill per part del director.

- **OwnClowd disk and Calendar:** Calendaris compartits, gestió de temps i compartició de documents, per tal de realitzar la comunicació entre les diferents parts implicades en el projecte, veure l'estat del projecte i realitzar un control de temps.
- **Asana:** A mesura que el projecte augmenti de complexitat, es possible que sigui necessari moure la gestió de temps a una solució més especialitzada com es Asana. Gràcies a la flexibilitat del sistema elegit, es molt simple i immediat el trasllat d'un sistema a un altre, per tant ens permet buscar el que millor s'adapti a les nostres necessitats en cada etapa del projecte.

## 3.4 Descripció de les tasques

### 3.4.1 Duració del projecte

El projecte té una duració limitada entre el mes de Febrer i el mes de Juliol de 2017, la part d'elaboració ha d'estar finalitzada a mitjans de Juny per a la seva distribució a la xarxa de sensors, i el seu període de proves s'estendrà més enllà de l'extensió del projecte durant la campanya de presa de dades a l'estiu de 2017. El projecte com a tal, serà finalitzat durant el Juny per a la seva presentació a la Facultat d'Informàtica de Barcelona aquell mateix mes.

### 3.4.2 Tasques

La planificació del projecte prèvia a la seva realització es la següent:

El projecte estarà dividit en diferents tasques i subtasques que queden definides per al seva duració i en el Gantt adjuntat mes endavant. La duració prevista per a cada tasca es descriu a la Taula .

TASCA	HORES
<b>1. Diseny i planificació de la infraestructura i les seves funcions</b>	130
1.1 Definir el projecte (GEP)	80
1.2 Establir les funcionalitats de l'aplicació	15
1.3 Definir quin programari s'haurà d'implementar i les seves funcionalitats	5
1.4 Disseny gràfic de la interfície d'usuari	30
<b>2. Implementació del software del servidor</b>	290
2.1 Scripts de control de sessions	10
2.2 Scripts de comunicació	10
2.3 Muntatge i posada a punt del servidor físic i la xarxa	20
2.4 Scripts d'inicialització de serveis	10
2.5 Posada a punt del Servidor Web	20
2.6 Estudi sobre programació web	20
2.7 Programació en python del back-end de control	70
2.8 Programació en python del servei web	45
2.9 Estudi sobre bases de dades	10
2.10 Disseny i creació de la base de dades	10
2.11 Programació del front-end bàsic	35
2.12 Programació del front-end avançat (terminal web)	10
2.13 Realització de tests i millores	20
<b>3. Implementació del software dels Sensors</b>	60
3.1 Scripts d'auto-control dels sensors	10
3.2 Scripts de comunicació i manteniment de la sessió	5
3.3 Testing de la comunicació	5
3.4 Modificació dels scripts d'instal·lació per a incloure el nou software	5
3.5 Instal·lació massiva del nou software a tots els sensors de la xarxa	20
3.6 Testing de la xarxa de sensors	15
<b>4. Testing amb els usuaris finals</b>	20
4.1 Prova pilot amb un nombre d'usuaris reduït	5
4.2 Realització d'un manual d'usuari	10
4.3 Prova final d'us amb usuaris dels diferents sectors implicats	5
<b>5. Documentació</b>	90
5.1 Escriure documentació final	60
5.2 Preparar presentació	30
<b>TOTAL:</b>	590

### 3.4.3 Dependències

Al Gantt adjunt, es poden visualitzar les diferents dependències. A continuació alguns aclariments:

Al Gantt es pot apreciar que algunes tasques es troben planificades per després de la finalització del projecte. Son totes les tasques de desplegament que es realitzaran durant la campanya d'estiu i que hem volgut incloure en el document per tenir una visió global de tota la duració del projecte.

Un coll d'ampolla important en el projecte es la preparació del servidor en el que s'allotjarà tot el projecte durant el seu desenvolupament i funcionament, per aquest motiu, es una de les primeres tasques que es realitzaran.

Finalment, la resta del projecte es divideix en tres blocs relativament independents entre ells: La programació del software dels sensors, la programació del software que usaran els usuaris finals, i la programació del servidor que realitzarà les comunicacions. Existeixen algunes dependències entre aquestes tasques, però la definició d'una interfície durant la fase de planificació, permetrà ignorar-les en gran mesura.

### 3.4.4 Recursos

Per a la realització del projecte, definim els recursos que s'usaran. En quant a recursos humans, el projecte serà portat a terme per una persona amb el suport dels voluntaris i operadors del projecte. Pel que fa a la resta de recursos, els podem diferenciar entre Hardware i Software:

#### Eines Hardware

- **40 Nodes CAPTOR:** Sistemes de sensors desenvolupats pel projecte, seran usats tant de forma directa com de forma remota
- **Lenovo U430p amb sistema operatiu Linux Mint:** Ordinador personal en el que es portarà a terme part del projecte
- **Ordinador de sobretaula Dell Optiplex amb sistema operatiu Linux Mint:** Ordinador proporcionat per el departament per tal de ser usat en el desenvolupament del projecte
- **Maquina Virtual (Pendent d'especificar):**Proporcionada per la universitat per allotjar el servidor del projecte.

	Nom	Durada	Gane 2017							Febrer 2017							Març 2017							Apol 2017							Maig 2017							Juny 2017							Juli 2017							Agos 2017							Setembre 2017						
			1	8	15	22	29	5	12	19	26	5	12	19	26	2	9	16	23	30	7	14	21	28	4	11	18	25	2	9	16	23	30	6	13	20	27	3	10	17	24																								
1	Definir el projecte (CEP)	80h																																																															
2	Establir les funcionalitats de l'aplicació	15h																																																															
3	Definir quin programari s'haurà d'implementar i les seves funcionalitgh																																																																
4	Disseny gràfic de la interfície d'usuari	30h																																																															
5	Scripts de control de sessions	10h																																																															
6	Scripts de comunicació	10h																																																															
7	Muntatge i posada a punt del servidor físic i la xarxa	20h																																																															
8	Scripts d'instal·lació de serveis	10h																																																															
9	Posada a punt del Servidor Web	20h																																																															
10	Estudi sobre programació web	20h																																																															
11	Programació en python del back-end de control	70h																																																															
12	Programació en python del servei web	45h																																																															
13	Estudi sobre bases de dades	10h																																																															
14	Disseny i creació de la base de dades	10h																																																															
15	Programació del front-end bàsic	35h																																																															
16	Programació del front-end avançat (terminal web)	10h																																																															
17	Realització de tests i millores	20h																																																															
18	Scripts d'auto-control dels sensors	10h																																																															
19	Scripts de comunicació i manteniment de la sessió	5h																																																															
20	Testing de la comunicació	5h																																																															
21	Modificació dels scripts d'instal·lació per a incloure el nou software	5h																																																															
22	Instal·lació massiva del nou software a tots els sensors de la xarxa	20h																																																															
23	Testing de la xarxa de sensors	15h																																																															
24	Prova pilot amb un nombre d'usuaris reduït	5h																																																															
25	Realització d'un manual d'usuari	10h																																																															
26	Prova final d'us amb usuaris dels diferents sectors implicats	5h																																																															
27	Escriture documentació final	60h																																																															
28	Preparar presentació	30h																																																															



### Eines Software

- **Git:** Software per gestionar versions del projecte
- **LibreOffice:** Suite Office per realitzar fulls de calcul i altres aplicacions necessàries per al projecte
- **GIMP:** Editor d'imatges, util per a la fase de disseny
- **LaTeX:** Escriure i compilar aquesta documentació
- **Ganttter:** Realitzar esquemes de Gantt i comprovar l'estat del projecte
- **Geddit:** Editor de text

## 3.5 Valoració d'alternatives i pla d'acció

### 3.5.1 Riscos

A continuació es valoraran els riscos que poden aparèixer durant el desenvolupament del projecte.

Durant la Fase d'especificació L'Únic risc important es en la tasca de disseny de la aplicació de l'usuari ja que els coneixements de disseny gràfic no son professionals.

Pel què fa al desenvolupament del software per al servidor, els riscos contemplats son: Baix coneixement de desenvolupament web, aprendre les tecnologies necessàries pot allargar el temps necessari per a una tasca; Retroacció, si els usuaris finals volen moltes modificacions, es pot allargar la tasca; Testeig, si en aquesta fase es trobessin molts errors, es podria allargar el desenvolupament.

A l'apartat del desenvolupament per als sensors es on trobem el risc mes gran del projecte: Les limitacions del hardware bàsic del sensor podrien no permetre portar a terme el desenvolupament previst.

No es valoren riscos notables a l'hora de documentar i presentar.

### 3.5.2 Alternatives i pla d'acció

Per a cada un dels riscos anteriors, es descriu un pla d'acció amb les alternatives per superar-ho:

Ja que L'únic risc durant la fase d'especificació son petites desviacions en la qualitat del producte final, no es precisa d'un pla d'acció important.

Pel que fa als riscos contemplats en el desenvolupament de software en el servidor, si al final resulta molt complicat usar les tecnologies previstes, el projecte s'haurà de dur a terme amb tecnologies mes senzilles de les que tinguem mes familiaritat, sacrificant qualitat del producte final. l'altre gran risc era una llista de canvis creixent degut a la retroacció dels usuaris finals, en aquest cas s'hauran d'analitzar les propostes i decidir quines poden ser implementades, quines no calen i quines es poden deixar per a un futur.

Finalment, el risc de la capacitat dels sensors s'haurà de salvar usant una tecnologia mes especifica (la solució no serà tant general com la prevista) que si funcioni en els limitats sensors.

Les alternatives elegides evitaran sempre al màxim allargar el projecte, ja que treballem amb uns períodes molt limitats per unes dates preestablertes de l'inici de la campanya d'estiu del funcionament dels sensors. Això significa que els costos d'haver d'usar una alternativa son el quantitat de característiques i qualitat del producte final (retallant el que no sigui possible) enlloc de temps.

## 3.6 Identificació y estimació dels costos

Degut a que les tasques tenen uns costos derivats, en recursos humans, Hardware, Software i altres indirectes, es important realitzar una aproximació de costos del projecte:

### 3.6.1 Recursos humans

El projecte serà realitzat per una sola persona que s'ocuparà de tots els rols del projecte. A continuació es detallen les hores invertides en cada rol i el seu preu (A partir de la descripció de tàsques anterior).

Fase	Rol				Cost estimat
	Cap de projecte	Dissenyador	Programador	Tester	
Disseny i planificació de la infraestructura i les seves funcions	80h	50h	0h	0h	4.700,00 €
Implementació del software del servidor	30h	0h	240h	20h	7.200,00 €
Implementació del software dels Sensors	0h	0h	40h	20h	1.400,00 €
Testing amb els usuaris finals	10h	0h	0h	10h	600,00 €
Documentació	90h	0h	0h	0h	3.600,00 €
<b>Total</b>	<b>210h</b>	<b>50h</b>	<b>280h</b>	<b>50h</b>	<b>17.500,00 €</b>

	Rol			
	Cap de projecte	Dissenyador	Programador	Tester
<b>Cost per hora</b>	40€/h	30€/h	25€/h	20€/h

### 3.6.2 Hardware

A part dels recursos humans, aquest projecte té uns costos associats al hardware que s'utilitzarà. A continuació es llisten els diferents elements que s'usaran i la seva amortització en el projecte. Per calcular l'amortització, s'ha pres com a referència un any de 250 dies laborables de 8 hores, la màquina virtual i els sensors funcionaran les 24 hores del dia. Es considera també que el 20% del temps es treballarà en el portàtil Lenovo i la resta en el sobretaula Dell :

Producte	Preu	Unitats	Vida útil	Amortització/h	Hores	Amortització
Dell Optiplex7040	600,00 €	1	4 anys	0,075€/h	472	35,40 €
Monitor Dell 24 polzades	320,00 €	1	4 anys	0,04€/h	472	18,88 €
Lenovo IdeaPad U430p	700,00 €	1	4 anys	0,0875€/h	118	10,33 €
Maquina virtual	120,00 €	1	1 any	0,014€/h	4320	60,48 €
Node CAPTOR	300,00 €	5	2 anys	0,085€/h	200	17,00 €
<b>Total</b>						<b>142,09 €</b>

Tot i que la xarxa està formada per molts més sensors, durant el període de desenvolupament es treballarà sobre 5 sensors en proves que són els que s'han especificat a la taula.

### 3.6.3 Software i llicències

Com a objectiu secundari i personal del projecte, s'intentarà utilitzar exclusivament productes de Software lliure, això redueix (fins a eliminar) completament els costos en Software i llicències.

L'únic Software no lliure que s'ha utilitzat ha sigut Ganttter per a requeriments de l'assignatura de GEP, aquest Software es gratuït d'usar i per tant no suposa un cost extra.

### 3.6.4 Altres costos

Un cop calculats els costos associats amb recursos humans i Hardware, hem de tenir en compte uns costos addicionals:

- **Costos Indirectes:** En aquest projecte tenim en compte dos costos indirectes:  
El primer es la oficina, que te capacitat per 6 persones i un cost aproximat de 600€/mes. Per tant, els costos derivats de la oficina son de 100€/mes durant 6 mesos, **600€**. Aquesta oficina es troba localitzada a una universitat publica i s'incluen els costos de llum, aigua i internet.  
El segon cost indirecte es el transport. Dos desplaçaments setmanals a Barcelona son 50€al mes (T-10 6 zones), durant 6 mesos: **300€**. Els desplaçaments dins la ciutat es realitzen amb una Bicicleta amb un cost aproximat de desgast i manteniment de **20€** durant els 6 mesos. El cost total dels costos indirectes es de **920€**.
- **Contingència:** La contingència cobreix les possibles variacions en la planificació temporal i costos del projecte (Veure apartat "Riscos"). Aquest projecte té una contingència fixada del 15% dels costos directes i indirectes del projecte. Per tant de **2784,30€**.
- **Imprevistos:** Finalment, sempre poden sorgir imprevistos. El principal imprevist que pot sorgir es troba a la màquina virtual, ja que pot ser que les característiques de les màquines virtuals ofertes pel DAC es quedin curtes i sigui necessari llogar el servei extern. Per tant es contemplen fins a **100€** (aproximadament 15€al mes de lloguer) per a aquest possible imprevist.

### 3.6.5 Resum de costos

Recursos Humans	Hardware	Software	Indirectes	Contingència	Imprevistos	Total
17.500,00 €	142,09 €	0,00 €	920,00 €	2.784,30 €	100,00 €	<b>21.446,39 €</b>

## 3.7 Control de gestió

Els costos seran controlats durant la reunió setmanal amb el cap de projecte, on s'avaluarà el progrés de les diferents tasques.

Al finalitzar cada tasca es compararà el temps i recursos invertits amb els previstos i es realitzarà un estudi de les variacions detectades per tal de prendre mesures i evitar possibles desviacions majors. El pressupost ha sigut dissenyat amb suficient marge per cobrir la majoria de desviacions, però s'haurà de revisar després de finalitzar cada grup de tasques per evitar que passi del 15% previst en el fons de contingència.

## 3.8 Desviacions de la planificació

Durant la realització del projecte, han sorgit diversos motius que han obligat a modificar la planificació temporal.

Ja que la realització del projecte s'efectuava sobre d'un treball anterior (disseny, programació i construcció dels nous models de sensors CAPTOR), el projecte no es podia iniciar completament fins que s'haguessin completat aquestes tasques. Per tant, una bona part del gruix del treball es va haver de desplaçar endavant en el temps per tal de poder acomodar en la planificació tot el temps dedicat a elles. Aquest desplaçament però, s'ha pogut paliar dedicant hores extres a les previstes en les setmanes posteriors a la finalització dels CAPTORS. A data d'avui, s'ha pogut atrapar la planificació temporal.

## 3.9 Modificacions sobre el plantejament inicial

Com hem explicat a la secció anterior, es va dedicar una bona part del temps planificat pel treball de final de grau a dissenyar, millorar i construir la nova generació de sensors CAPTOR. Ja que es considera que aquestes tasques s'adeqüen al context del treball, s'afegiran a la memòria del projecte.

En resum, les tasques extra que s'han realitzat son:

- **Anàlisi d'errors i proposta de millora:** Partint del treball existent, s'han analitzat els problemes existents a les antigues generacions de sensors CAPTOR i s'han proposat possibles fonts d'aquests problemes. S'han proposat solucions i millores.
- **Disseny de placa integrada:** Ja que la causa més probable d'error era el mètode provisional de construcció del circuit lògic del captor, es proposa realitzar una placa amb els components soldats. Aquesta placa es dissenya amb software CAD professional, pensant en la seva manufacturació.

- **Realització manual de prototipus:** Es prova el funcionament del disseny realitzant un prototipus manualment amb placa de coure i mètode de transfer de tòner.
- **Avaluació de la solució:** Es prova el prototip i es decideix que la solució funciona, es decideix produir les plaques de forma professional.
- **Adaptació i construcció de les plaques:** S'adapten les plaques per a la seva fabricació. Un cop rebudes, es solden tots els components de forma manual.
- **Construcció i programació dels CAPTOR nous:** Es construeixen, programen i testejen els nodes CAPTOR nous. El resultat dels tests es positiu.

# Capítol 4

## Sostenibilitat

### 4.1 Sostenibilitat i compromís social

Resumim l'impacte del projecte en la matriu de sostenibilitat:

	PPP	Vida Útil	Riscos
Ambiental	Consum del disseny	Petjada ecològica	Riscos Ambientals
	7/10	17/20	-3
Econòmic	Factura	Pla de viabilitat	Riscos Econòmics
	8/10	17/20	-4
Social	Impacte Personal	Impacte Social	Riscos Socials
	9/10	18/20	0
Rang Sostenibilitat	24/30	52/60	-7
	69/90		

#### 4.1.1 Impacte ambiental

Per a la realització del projecte es reutilitzen el màxim de dispositius possibles per tal d'evitar el cost ecològic que comprar dispositius nous comportaria. Es tracta d'ordinadors relativament nous amb uns consums baixos. Els servidors es troben funcionant en CPDs especialitzats i molt eficients, que redueix el consum respecte si s'utilitzés un ordinador convencional.

Finalment, degut a la orientació ecologista del projecte, i a una de les seves finalitats principals: reduir el nombre de desplaçaments físics (i les emissions que comporten) fins als sensors permetent que puguin ser controlats i mantinguts de forma remota, el projecte té un impacte ambiental positiu en quant a les emissions que s'eliminaran gràcies a la

implantació dels sistema dissenyat i implementat.

**Impacte ambiental en la societat:** Ja que aquest projecte forma part d'una campanya científica per conscienciar i obtenir dades que permetin la reducció de la contaminació atmosfèrica per ozó, les seves implicacions ecològiques son evidents. El projecte ajuda a tenir un millor coneixement sobre la contaminació ambiental.

#### 4.1.2 Impacte econòmic

Ja que el sistema actual de control i manteniment dels sensors representa un cost de recursos humans molt elevat (La implantació del sistema pot reduir a mitja hora una tasca que actualment requereix fins a un dia en desplaçaments dels responsables del manteniment dels sensors), i al ser els recursos humans la part mes costosa actualment, la implantació del projecte permetrà reduir considerablement el cost que suposa aquesta diferència de temps.

Pel que fa als costos de Hardware, al ser difícil usar alternatives mes econòmiques al estar usant Hardware específic i especialitzat, el cost del projecte pel que fa a Hardware es el màxim d'ajustat possible.

**Impacte econòmic en la societat:** Un dels objectius del projecte en el que col·laborem és reduir l'impacte econòmic de la investigació de la contaminació de l'ozó investigant formes d'usar diversos sensors de baix cost per obtenir les mateixes dades que un sol sensor de molt alt cost, creiem que el projecte tindrà un impacte econòmic favorable en la societat, reduint els costos de l'investigació mentre es conserva la qualitat de les dades preses.

#### 4.1.3 Impacte Social

Aquest projecte aporta a nivell personal una major consciència sobre el problema de la contaminació ambiental i uns millors coneixements sobre com la tecnologia i les telecomunicacions poden ajudar a pal·liar-lo i ser una eina de millora de l'ambient. Especialment pel què fa a les possibilitat d'Internet com eina de conscienciació social i d'acostament de investigació científica a la ciutadania.

Finalment, el projecte participa en un projecte de codi obert i participatiu en millora constant i que podrà ser usat per a futurs projectes.

**Impacte social en la societat:** Ja que el projecte en el que participem busca la col·laboració de sectors de la societat que normalment no participen en àmbits científics, col·laborem en conscienciar i apropar l'investigació a la població general.



## Capítol 5

# Anàlisi i millora del node CAPTOR

### 5.1 Introducció

Durant l'estiu del 2016, la primera campanya de presa de dades dels sensors CAPTOR es va dur a terme en diverses localitzacions de Catalunya, Àustria i Itàlia. Les campanyes estan formades per dues fases de cal·libració seguides per una fase de presa de dades a la localització final.

Durant les fases de cal·libració, els sensors es posen a prop d'una estació de referència: En la primera fase de cal·libració, es calibren a la estació de referència del CSIC a Palau Reial, durant la segona fase, es calibren a una estació de referència propera al seu emplaçament definitiu. Aleshores es comparen les dades preses per els sensors amb les preses per aquestes estacions de referència (Dades que anomenarem "reals" degut a la alta exactitud que tenen) durant el mateix període de temps. Aquestes dades finalment s'utilitzen per ajustar les dades de la fase final i obtenir uns valors que s'acosten amb fiabilitat a les reals.

Durant la fase de cal·libració, però, es van detectar alguns problemes sense motiu aparent que feien que les dades preses no fossin útils per a la cal·libració dels sensors. El problema més notable era que sense aparent explicació, tots els sensors del node canviaven rang dinàmic, prenent dades fins a 100 vegades inferiors.

A la Figura 5.1, es pot veure aquest problema amb la cal·libració del captor 16002. Els valors en rosa de les dues primeres gràfiques (corresponents a la cal·libració 1 i la cal·libració 2 respectivament) són els corresponents a l'estació de referència, mentre que els altres, representen cada un dels sensors del node. L'eix vertical de l'esquerra representa els kOhms marcats pels sensors d'ozó, representatius per als sensors del node, l'eix vertical de la dreta són els mil·ligrams per metre cubic d'ozó detectats pel sensor de l'estació de

referència. Es normal que diferents sensors en el mateix node treballin en rangs dinàmics diferents, mentre el sensor es mantingui sempre dins del mateix rang.

Com es pot veure en la primera de les gràfiques, en un punt en concret, les dades preses pel sensors canvien totalment de rang dinàmic, passant d'assenyalar uns valors sobre els 800 kOhms a la primera cal·libració, a uns valors sobre els 200 KOhms a la segona cal·libració i a la presa de dades definitiva.

Al mantenir-se les dades entre la segona cal·libració i la presa final, encara poden ser útils, però al tenir en compte la primera cal·libració, amb el canvi de rang dinàmic, les dades preses perden el seu valor. A la figura 5.2, es pot veure com les dades de la primera cal·libració (fila superior) no segueixen la regressió esperada que les de la segona cal·libració (fila inferior) si que segueixen.

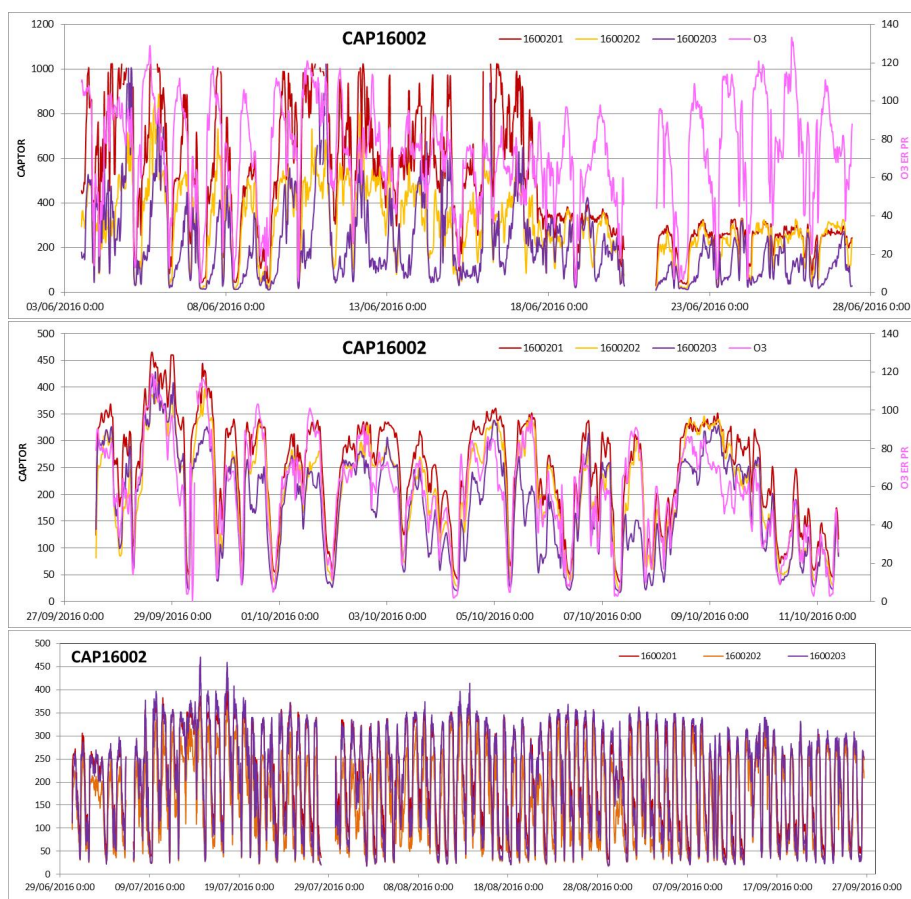


Figura 5.1: Calibració CAPTOR 16002

**Font:** Projecte CAPTOR

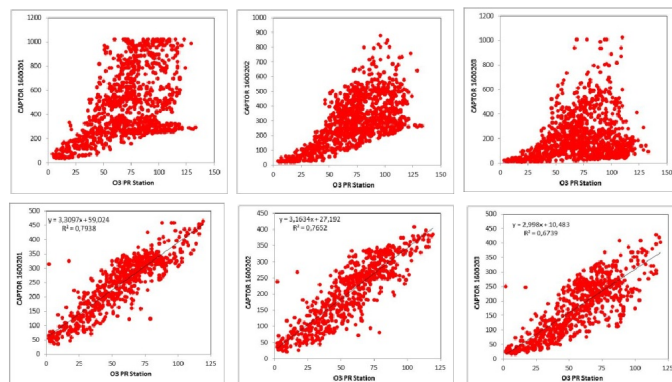


Figura 5.2: Calibració CAPTOR 16002

**Font:** Projecte CAPTOR

Aquest no es un error aïllat en un sol node, el mateix problema es pot veure en altres nodes: La figura 5.3 mostra el mateix efecte en el CAPTOR 16005. Que passa de mostrar uns valors sobre els 80 kOhms durant la fase de cal·libració, a mostrar uns valors sobre els 20 kOhms a la fase de presa de dades. En aquest cas, al no coincidir en cap de les fases, podem veure a la figura 5.4, que la cal·libració no es pot usar ja que no es poden relacionar els valors de la presa final amb els calibrats.

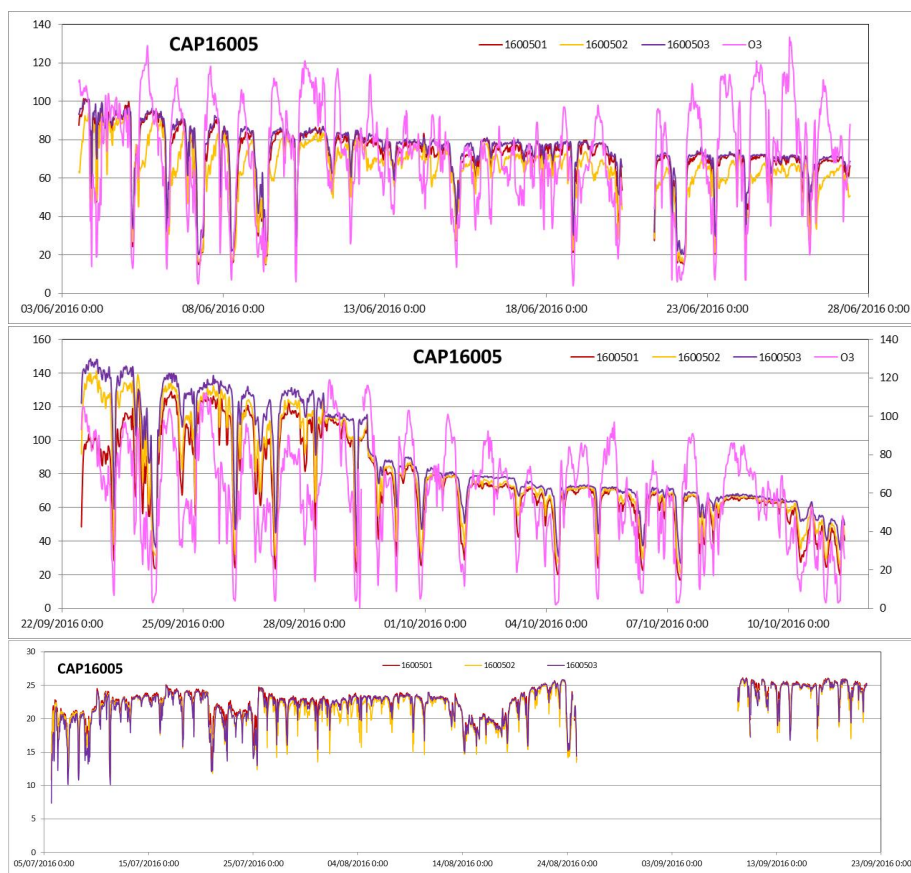


Figura 5.3: Calibració CAPTOR 16005

Font: Projecte CAPTOR

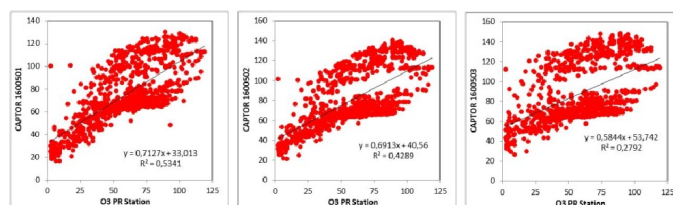


Figura 5.4: Calibració CAPTOR 16005

Font: Projecte CAPTOR

## 5.2 Objectius

Els objectius d'aquesta part del Treball de Final de Grau, es poden resumir en:

- Detectar el problema
- Proposar solució
- Dissenyar prototip de la placa
- Realitzar un prototip de la placa
- Efectuar una prova de la solució amb el prototip de la placa
- Preparar el disseny per a fabricació en sèrie
- Soldar i testejar totes les plaques
- Proposar millores estètiques i d'estabilitat
- Muntar tots els nodes CAPTOR per a la campanya 2017

## 5.3 Tecnologies

A continuació, descriuré les tecnologies i conceptes que s'han usat per a realitzar aquesta part:

### 5.3.1 Arduino

Arduino és una placa de desenvolupament de projectes electrònics, amb un microcontrolador de tipus Atmel AVR. La placa normalment disposa de pins E/S així com la majoria de components que el microcontrolador necessita per funcionar. En la majoria de models, tots els seus components son accessibles a través d'una llicència de hardware i software lliure. Disposa d'un entorn de desenvolupament en C++.

Els nodes CAPTOR utilitzen com a processador, un Arduino Yun (Com el de la figura 5.5). La peculiaritat d'aquest Arduino en concret es que a part del microprocessador tradicional Atmel dels Arduinos, porta també un petit processador MIPS que executa una distribució de Linux (OpenWRT, al que anomenen Linino). Això ens permet programar-lo com si es tractés d'un petit ordinador, amb molta mes potencia que un Arduino tradicional.

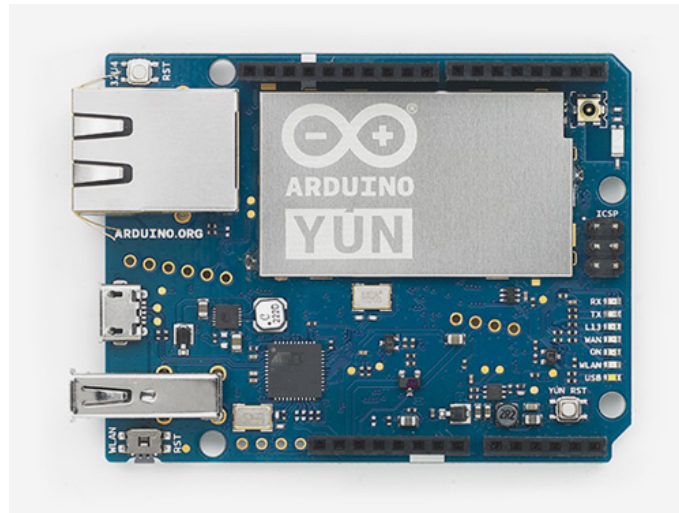


Figura 5.5: Arduino Yun

**Font:** Arduino Foundation

Per tant, la placa Arduino Yun disposa de dos entorns, un entorn purament Arduino, que s'encarrega de tractar amb les dades analògiques dels sensors, i un entorn Linux, que gràcies a ser més potent, s'usa per administrar l'emmagatzematge, la connectivitat, i totes les funcions extra de les que disposa aquest Arduino.

Els dos entorns, es comuniquen entre ells a través d'un Bridge controlat per un daemon, que mitjançant operacions de baix nivell, transmet les dades entre els dos entorns.

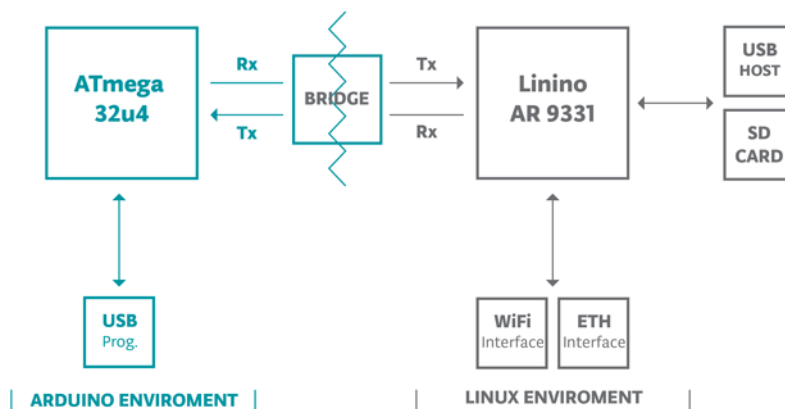


Figura 5.6: Bridge entre els dos entorns de l'Arduino Yun

**Font:** Arduino Foundation

### 5.3.2 OpenWRT



Figura 5.7: Logotip d'OpenWRT

**Font:** OpenWRT

OpenWRT és un sistema operatiu basat en Linux i pensat per a routers i altres dispositius encastats i de baix consum. OpenWRT està especialment orientat a dispositius de xarxa com són els routers, però en aquest cas, l'usem com a sistema operatiu de l'Arduino. L'Arduino porta una versió personalitzada d'OpenWRT anomenada Linino.

A efectes pràctics, OpenWRT és una distribució totalment funcional de Linux. En el nostre cas, el programarem usant Python, però és capaç de córrer la majoria de llenguatges de programació moderns.

### 5.3.3 EAGLE



Figura 5.8: Logotip d'EAGLE

**Font:** CadSoft

EAGLE és un software de disseny assistit per computador especialitzat en el disseny i planificació de plaques de circuit imprès (PCB). En el projecte s'usarà per al disseny del prototip i la placa definitiva.

S'utilitza la versió de la comunitat d'EAGLE que permet utilitzar un màxim de dues capes de placa, més que suficients per als dos usos que es realitzen durant el curs del treball.

### 5.3.4 Fabricació manual de PCB

Mètode de fabricació de PCB amb elements domèstics i de baix cost.

Qualsevol mètode casolà de fabricació de PCB es compon de 3 fases o elements:

1. **Emmascarar les traces:** S'utilitza algun sistema per a protegir la part de coure de la placa verge que vulguem mantenir en el disseny final, és a dir, traces, vies, pads, senyals de guia, punts de test i altres elements decoratius.
2. **Eliminació del coure no desitjat:** Amb un mètode usualment químic (tot i que també n hi ha de mecànics, molt menys precisos), s'elimina el coure de la placa verge no desitjat en la PCB final.
3. **Acabats finals:** Segons el resultat desitjat de la placa, es poden aplicar els acabats finals, com poden ser perforar forats per a components, perforar i unir dos capes a través de vies, vernisar el coure per protegir-lo i facilitar la soldadura de components, aplicar una serigrafia al costat dels components...

Pel que fa a la màscara, nosaltres usarem el mètode per transferència de tòner, que és un mètode molt barat que permet aconseguir unes plaques amb un acabat polit utilitzant material casolà i de molt baix cost. El mètode de transferència de tòner es pot veure en la figura 5.9 i és el següent:

1. S'imprimeix el disseny final de la PCB en negre utilitzant una impressora làser en un full de paper prim i setinat (per exemple un full de publicitat per correu es una bona opció). La impressió s'ha d'invertir per tal que un cop transferida quedi amb la orientació desitjada.
2. Es neteja molt be la placa de coure verge i es poleix lleugerament la seva superfície usant un fregall.
3. Es col·loca la impressió sobre la placa de forma que el tòner toqui al coure.
4. S'aplica escalfor al paper, per exemple amb una planxa per a roba, per tal de que la placa de coure s'escalfi i el tòner es desfaci i adhereixi al coure.
5. S'espera a que es refredi la placa i es submergeix en aigua fins que el paper es desprengui deixant el tòner transferit a la placa.

Finalment, eliminarem el coure restant amb el mètode mes barat, senzill i segur: Submergint la placa amb la mascara en una solució a parts iguals d'Àcid clorhídric i peròxid d'hidrogen, és a dir, Salfumant (en una concentració del 30% es suficient, la usada per netejar) i aigua oxigenada (10 vol de farmàcia funciona be). Això eliminara el coure que no es trobi protegit amb el tòner. Finalment, eixuguem be la placa per aturar la reacció, i eliminem el tòner amb acetona.





Figura 5.9: Proces d'aplicació de la màscara de tòner

Font: Imatges cortesia de Pinomelean

## 5.4 Desenvolupament

### 5.4.1 Proposta del problema

Degut a que els canvis de rang dinàmic no mostraven ser provocats per cap error de software, s'analitza el hardware i proposa la hipòtesi de que les resistències de càrrega dels sensors veuen el seu valor modificat amb el moviment del node. És a dir, que el contacte del cable, al no ser correcte o prou segur, es pot moure i modificar l'impedància del sistema, i per tant, modificar el seu rang dinàmic.

Aquesta hipòtesi es veu suportada pel fet de que els canvis es produeixen al moure o traslladar els nodes. A més, degut a la forma en la que els nodes estaven fabricats durant la campanya del 2016, en la que els components eren aguantats per una placa de prototip (tal i com es pot veure a la figura 5.10), els moviments eren una causa molt probable dels canvis en els contactes de les connexions en la placa i per tant, de lectures dels sensors totalment diferents.

Per tant, es proposa una solució: realitzar una placa de circuit imprès en lloc d'usar una placa de prototips. Creiem que la placa de circuit pot oferir molts punts positius al projecte, com són major estabilitat en les lectures, menys problemes causats pel moviment dels nodes, major facilitat de muntatge i aspecte més cuidat i professional.

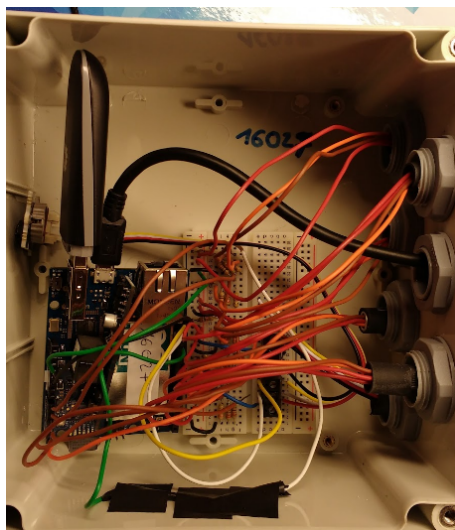


Figura 5.10: Connexions internes del CAPTOR durant la campanya del 2016

**Font:** Projecte CAPTOR

#### 5.4.2 Disseny de la solució

Per a dissenyar la placa, usarem EAGLE.

El primer pas a realitzar, és dibuixar l'esquema de connexions de la placa, ja que tota la informació de la que es disposa per saber com estan connectats els components de la placa son nodes ja muntats i la imatge de la figura 5.11, es col·loquen els components en el full de dibuix de la forma que es troben organitzats en l'imatge, i guiant-nos per les connexions mostrades i l'ajuda d'un multímetre amb capacitat de realitzar tests de continuïtat, deduïm l'esquema electrònic de la figura 5.12.

La raó de perquè aquest esquema es troba tant desorganitzat i amb un estil poc ortodox és doncs degut al fet que ha estat realitzat mitjançant enginyeria inversa de nodes ja muntats i no te més propòsit que indicar al software de disseny de plaques quines són les connexions entre els components.

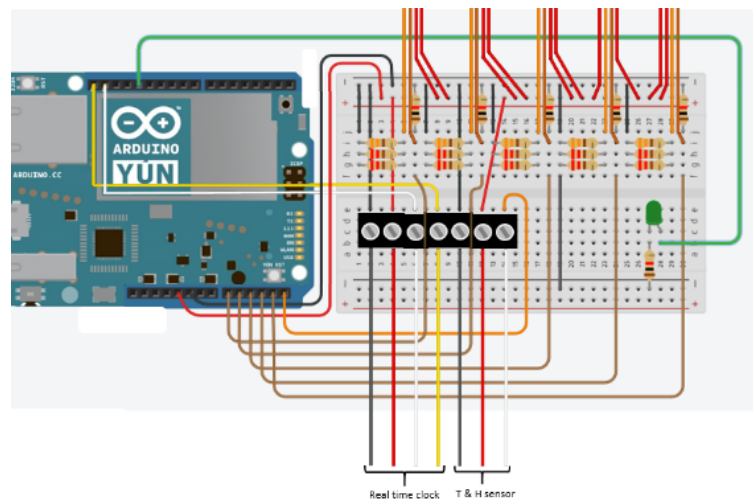


Figura 5.11: Diagrama de connexions del que es parteix per realitzar l'esquema

**Font:** Projecte CAPTOR

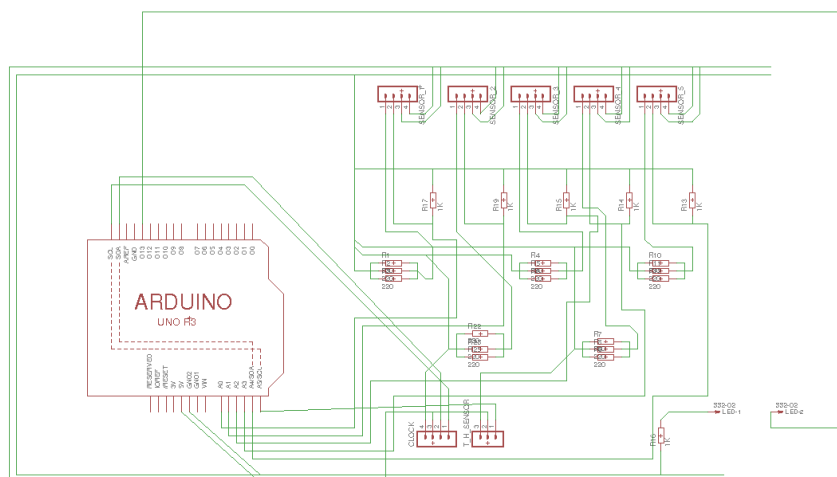


Figura 5.12: Diagrama auxiliar de les connexions

### 5.4.3 Fabricació del prototip

Per tal de poder provar si el us d'una placa soldada millora els resultats, es decideix realitzar un prototip de la placa de forma casolana.

Utilitzant com a guia l'esquema mencionat, es dissenya la placa. Es col·loquen els components de forma organitzada per facilitar el seu muntatge i es decideixen utilitzar pins femella de separació estàndard per tal de facilitar el canvi de sensors que puguin funcionar malament. També es decideixen usar el mateix tipus de pins per a elements externs al Arduino que s'hagin de connectar: El LED d'indicació i el rellotge de temps real.

Finalment, amb ajuda puntual del Autorouter (Una ajuda del software per passar traces entre components), es dissenya un mapa de traces per al prototip. Aquestes capes usaran una sola cara de la placa per facilitar la seva fabricació casolana.

El resultat i disseny del prototip és el que es pot veure a la figura 5.13. En vermell les traces, en verd els pads de soldadura, i en gris una senzilla serigrafia per tal de facilitar la col·locació definitiva dels elements en la placa.

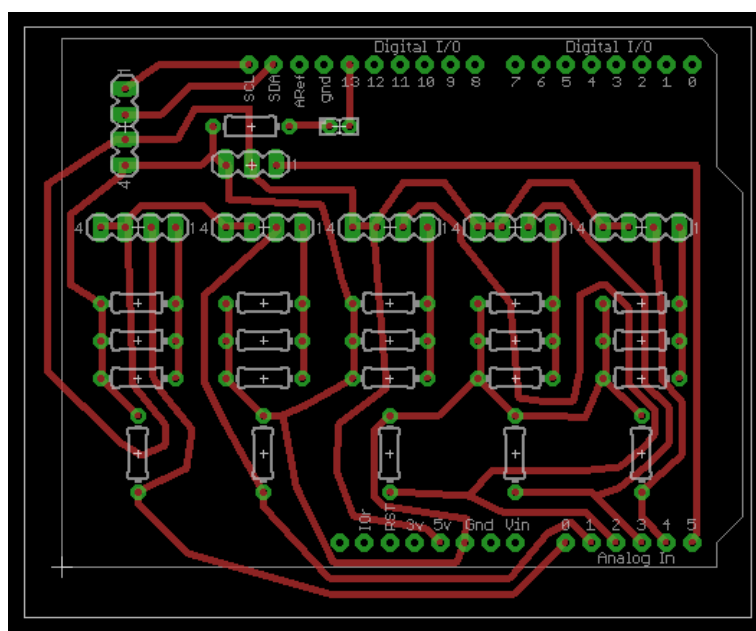


Figura 5.13: Diagrama final del prototip

A continuació es fabrica el prototip per mètode del tòner, de la forma descrita al apartat anterior. A la imatge de la figura 5.14 es pot veure el resultat de la placa després de gravar-la i de netejar el tòner. Finalment apliquem pel cantó dels components, amb el mateix

mètode de tòner i planxa, una senzilla serigrafia per tal d'assistir en el muntatge dels components.



Figura 5.14: Placa finalitzada i serigrafia

Finalment, es solden els components a la placa, i es decideix fabricar una segona copia per tal de poder tenir més dades en els tests. Es decideix soldar uns connectors mascle el costat del coure per tal de simplificar la connexió del prototip al Arduino: simplement es connecta a sobre directament, fent coincidir tots els pins.



Figura 5.15: Prototips finals

#### 5.4.4 Test del prototip

Es seleccionen el captor 16021 i 16022 com a captors de proves per tal de testejar la solució. Alguns tests que passen aquests CAPTOR són:

- Diverses rondes de cal·libració a la estació de referència de palau reial.
- Trasllats i moviments

- Cops i sacsejades brusques
- Reinicis forçats

Com veurem en més detall al apartat d'anàlisi de resultats, sembla que el fet d'usar les plaques és positiu en la qualitat de les dades preses pel CAPTOR.

### 5.4.5 Disseny i fabricació de la placa final

Per la placa que finalment s'encarrega a fabricar, es decideix canviar lleugerament el disseny. S'usaran dues capes per tal de reduir el factor de mida de la placa i facilitar el muntatge dels pins mascle que la connecten al Arduino, es canviarà la forma de la placa per evitar errors de connexió al revés, i s'afegiran detalls gràfics a la serigrafia, amb informació sobre el projecte i sobre el muntatge dels components connectables a la placa.

El disseny definitiu que s'encarrega fabricar és el de la imatge de la figura 5.16, en vermell i blau les traces en les dues capes, en verd les vies i pads, i en gris la serigrafia. S'ha inclòs en la serigrafia un rectangle en blanc per escriure el codi de cada CAPTOR i facilitar-ne el seu reconeixement ràpid.

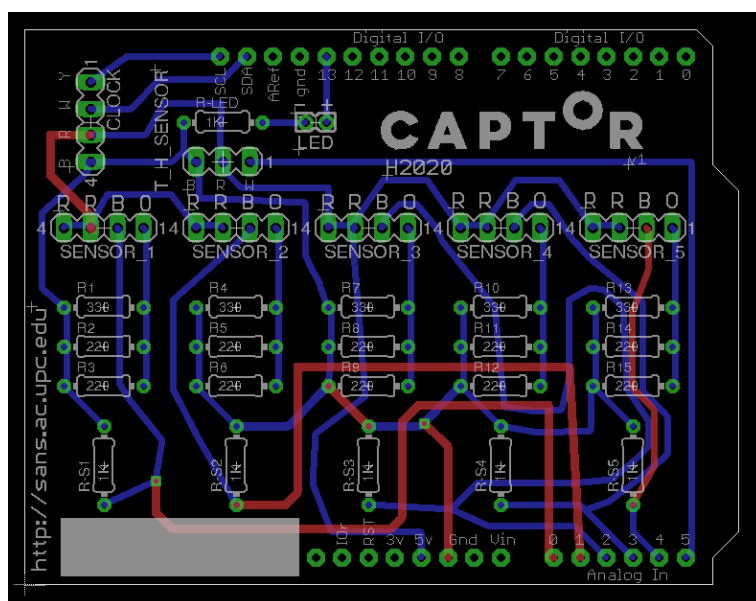


Figura 5.16: Diagrama final de la placa del CAPTOR

Finalment, encarreguem la fabricació d'aquesta placa. El resultat final és el de la imatge de la figura 5.17. Els components no venen soldats a la placa, i es dediquen unes setmanes al muntatge de plaques.



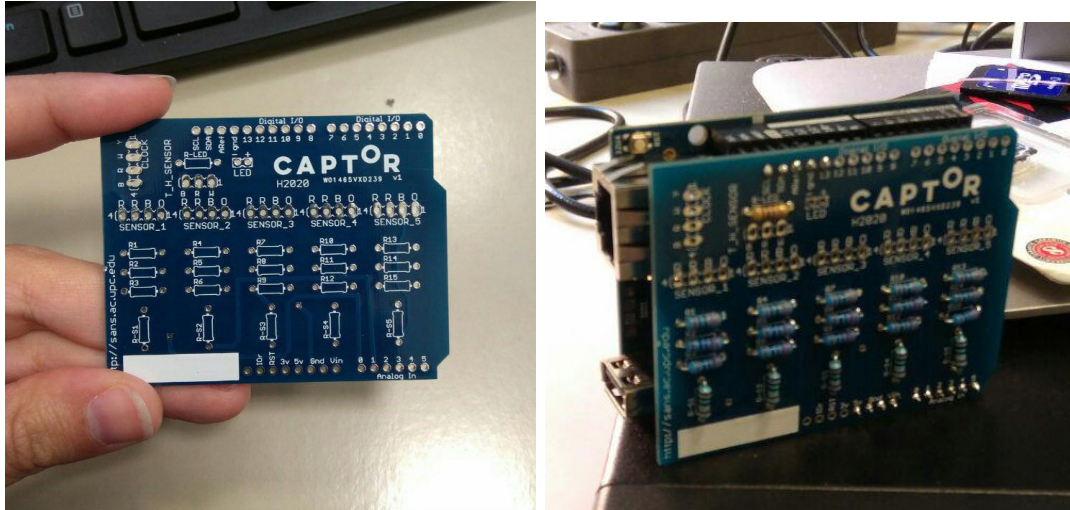


Figura 5.17: Placa final per als CAPTOR de la campanya 2017

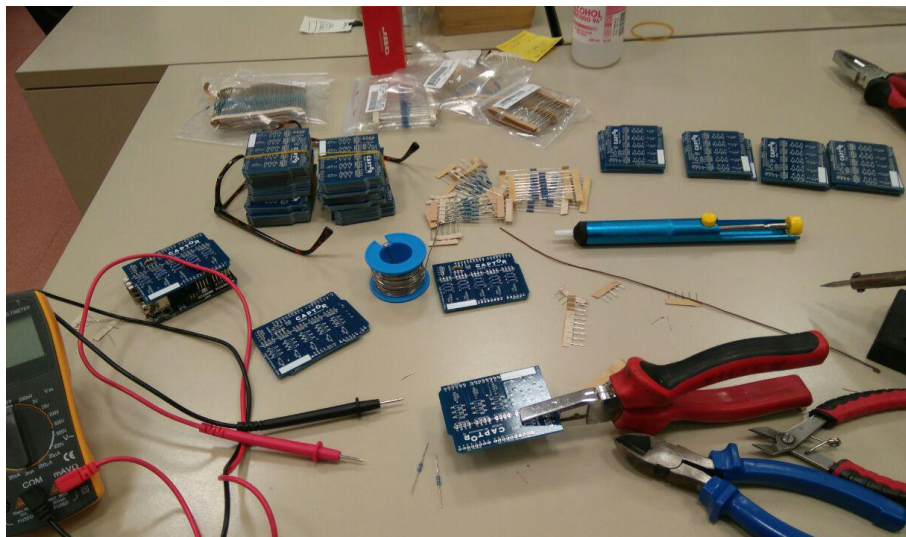


Figura 5.18: Procés de muntatge dels nodes CAPTOR

## 5.5 Anàlisi de resultats

Gràcies als tests realitzats amb els prototips 16021 i 16022, i amb les dades obtingudes durant les primeres setmanes d'aquesta campanya, es pot veure que les plaques han tingut un impacte positiu en la qualitat de les dades, i que no s'han tornat a repetir els canvis de rang dinàmic que s'observaven en la campanya anterior.

En la figura 5.19 es mostra per exemple els resultats de la segona cal·libració del captor 17013, a Manlleu. En el primer diagrama, en taronja veiem les dades preses per l'estació de referència, i en la resta de colors, els diferents sensors del CAPTOR. Es pot veure que realment les dades tenen una correlació molt notable, i que no es veuen salts de rang dinàmic com passava en els CAPTOR de la temporada anterior.

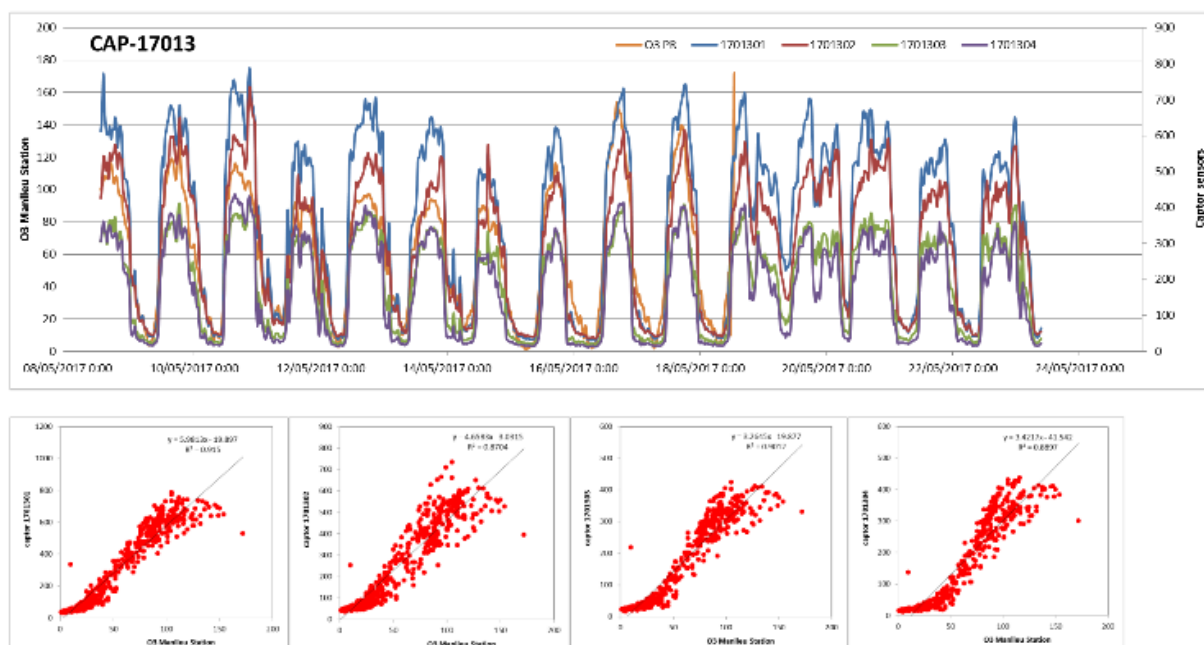


Figura 5.19: Segona cal·libració del CAPTOR 17013

**Font:** Projecte CAPTOR



Finalment, l'aspecte intern del node és molt més cuidat i estable: Al estar tot subjectat no es veu afectat per cops i moviments com sí ho feien els nodes de la campanya anterior:

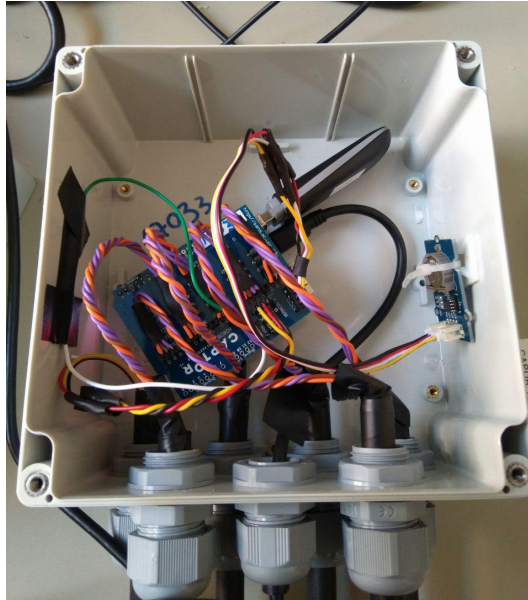


Figura 5.20: Node final

**Font:** Projecte CAPTOR



# Capítol 6

## Interfície web de gestió

### 6.1 introducció

Actualment els administradors i responsables del manteniment dels nodes CAPTOR utilitzen una quantitat d'eines no-específiques i disperses per realitzar el manteniment dels nodes. A més aquestes eines en molts casos necessiten un procés d'instal·lació, cosa que obliga als responsables a treballar sempre des del mateix ordinador. Això no permet per exemple, utilitzar un dispositiu mòbil per realitzar manteniment i controlar els nodes en les seves pròpies localitzacions.

Per tant, es decideix implementar una eina concreta per al projecte CAPTOR, que ajunti en una de sola les eines necessàries per portar un registre de la informació de tots els nodes, l'estatus de cada un d'ells, i finalment, la comunicació amb aquests nodes de forma remota sense necessitat de desplaçar-se físicament o connectar-se per VPN a la xarxa on estan connectats els nodes.

Aquesta eina a més, serà en forma de pàgina web, per tal de que sigui accessible des de qualsevol dispositiu de forma immediata i sense necessitat d'instal·lació de cap software.

### 6.2 Objectius

Els objectius d'aquesta part del Treball de Final de Grau, es poden resumir en:

- Crear un servei web accessible des de qualsevol navegador i que implementi les següents funcions:
  - Permetre afegir, editar i eliminar informació sobre els nodes.
  - Permetre la monitorització en temps real dels nodes.

- Permetre la connexió remota per SSH amb els nodes de forma directa i des del mateix navegador.
- Aconseguir que el servei web sigui estable i segur.

## 6.3 Tecnologies

### 6.3.1 Python



Figura 6.1: Logotip Python

Python és un llenguatge de programació interpretat que posa molt ènfasis en tenir una sintàxi altament llegible.

El backend complet de la pàgina web ha sigut programat en Python. S'ha elegit aquest llenguatge de programació degut a que la resta de programari del projecte CAPTOR està programat també en Python. Per tant, s'ha decidit usar Python per mantenir la cohesió entre els projectes.

Una de les característiques més beneficioses de Python per aquest projecte és la seva facilitat de paralelització. Gràcies a ella, la nostra pàgina web podrà servir de forma concurrent més d'una petició inclús si una d'elles es troba realitzant alguna tasca bloquejant. Algunes de la pàgines del nostre sistema, com veurem més endavant, mantenen el servidor realitzant una tasca de forma continuada un cop servida la pàgina fins que el client abandona la connexió, volem que mentre el servidor està realitzant aquesta tasca bloquejant, pugui seguir servint les peticions dels altres possibles clients. Per això, destinarem més d'un procés a servir les pàgines.

### 6.3.2 Flask

Flask és un microframework de programació web escrit en Python. Flask permet crear aplicacions web ràpidament i amb un nombre mínim de línies de còdi, a més, al ser



Figura 6.2: Logotip Flask

tansols un microframework, permet una enorme flexibilitat, deixant màxima llibertat al programador per estructurar l'aplicació beb segons les seves necessitats.

Flask incorpora integració amb el motor de plantilles HTML Jinja2, que ens permetrà generar HTML de forma dinàmica.

L'estructura bàsica d'una aplicació en Flask es extremadament senzilla:

---

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route("/")
5 def hello():
6     return "Hola Mundo"
7
8 if __name__ == "__main__":
9     app.run()
```

---

Codi 1: Estructura bàsica d'una aplicació web en Flask

En aquest exemple, al accedir al home de l'aplicació des d'un navegador web corrent, es retornarà una pàgina amb una salutació.

La nostre aplicació web estarà programada en Flask, que ens facilitarà la creació de totes les pàgines que necessitem mostrar. Ja que la nostra aplicació estarà formada bàsicament per fitxes o plantilles de cada CAPTOR amb estructura igual però contingut diferent, Flask és una molt bona opció ja que està especialment orientat a aquest tipus d'us.

### 6.3.3 HTML

Les pàgines que mostrarem al navegador seran programades en HTML, Així com els formularis per entrar dades al sistema.

Com hem mencionat, Flask inclou un motor de plantilles HTML anomenat jinja2. Aquest potent motor, ens permet generar de forma dinàmica les pàgines HTML que retornarem al navegador del client usant un llenguatge de programació basat en javascript amb el que podem incloure valors de dades passades des del backend en Python, inclús navegant estructures d'objectes compostos.

Com a exemple de pàgina senzilla que usa Jinja2:

---

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>{{ variable|escape }}</title>
5   </head>
6   <body>
7     {% for item in item_list %}
8       {{ item }}{% if not loop.last %},{% endif %}
9     {% endfor %}
10  </body>
11 </html>
```

---

Codi 2: Exemple de Jinja2

En aquest exemple podem veure com podem incloure fàcilment valors de variables en el codi encapsulant-les dins de `{{ }}`. A més, es poden usar estructures de control per repetir o incloure parts del codi, encapsulant-les dins de `{% %}`.

Per ajudarnos en el format de la pàgina, i simplificar la creació d'un disseny més cuidat, usarem Bootstrap, que és simplement una estructura de classes CSS amb un disseny agradable.

### 6.3.4 Javascript

El codi que s'executarà en la part del client, estarà programat en Javascript. Javascript és un conegut llenguatge d'scripting especialment popular per el seu ús en navegadors, es pot incloure fàcilment en l'HTML d'una pàgina web i permet executar en el navegador petites peces de codi.

A la pàgina a més usarem dues tecnologies derivades de Javascript: **AJAX** i **JQuery**:

- **AJAX:** Es una tècnica per tal de sol·licitar des del client, informació al servidor de forma asíncrona, sense necessitat de recarregar la pàgina.

- **JQuery:** És una famosa llibreria de Javascript per a facilitar la programació d'scripts d'una web, principalment scripts orientats a modificar l'HTML de la pàgina.

### 6.3.5 Server-Sent Events

Els Server-Sent Events (SSE) és un estandar relativament nou, estandaritzat en una W3C Recommendation del 2015[8].

Quan s'usa un Server-Sent Event, el navegador, gràcies a un script en el Javascript de la pàgina, es connecta a un "Stream" de dades provinent del Servidor. En el moment que el servidor actualitza les dades en aquell stream, el client veu el canvi i pot actuar en conseqüència.

Hem decidit usar SSE en el nostre projecte per 3 raons principals:

- **És una tecnologia nova i amb molt de futur:** Hem decidit experimentar amb aquesta nova tecnologia i per tant augmentar la quantitat de recursos i informació que corre sobre ella a internet, que és actualment molt limitada.
- **No és necessari polling:** Podem rebre dades del servidor sense necessitat de realitzar peticions periòdiques, cosa que multiplica el tràfic de i cap al servidor, ja que cada petició és una connexió TCP independent.
- **Disminueix la càrrega al client i al servidor:** A part de pel motiu anterior, el servidor pot enviar les dades al seu ritme i no existeix la possibilitat de que el client faci una segona petició abans de que el servidor hagi pogut acabar de respondre a la primera d'elles. A més, si hi ha més d'un client escoltant el mateix stream, el servidor no ha de realitzar feina extra.

A continuació veurem un exemple del funcionament d'un SSE, que veurem amb més detall al apartat de desenvolupament d'aquest capítol:

---

```
1 var targetContainer = document.getElementById("this-div");
2 var eventSource = new EventSource("/stream");
3
4 eventSource.onmessage = function(e) {
5     targetContainer.innerHTML = e.data;
6 };
```

---

Codi 3: Exemple d'SSE, Client side

---

```
1 @route("/stream")
2 def stream():
3     def eventStream():
4         while True:
5             # Poll data from the database
6             # and see if there's a new message
7             if len(messages) > len(previous_messages):
8                 yield "data:
9                     {}\n\n".format(messages[len(messages)-1]))"
10
11     return Response(eventStream(), mimetype="text/event-stream")
```

---

Codi 4: Exemple d'SSE, Server side

Podem veure com des del client, ens subscriuim a un EventSource que es troba a */stream* en el nostre servidor d'aplicacions, i que cada cop que rebí un missatge per aquest stream, canviarà el contingut d'un element de la pàgina. Per altre banda, des del servidor, podem veure com esta format un event: Cada línia ha de començar obligatòriament amb "data:" i finalitzar amb doble salt. Sense aquest format, el navegador no ho interpretarà com a part del stream.

### 6.3.6 NodeJS, WebSockets i Wetty



Figura 6.3: Logotip NodeJS

Una de les funcions del nostre servei és proporcionar un terminal online per a comunicar-se per ssh de forma segura amb el node CAPTOR sense necessitat d'Instal·lar res addicional al client, per tant, totes les funcionalitats seran a través del navegador.

Per dur a terme aquesta tasca, utilitzarem wetty[9], un petit emulador de terminal per a navegador, software lliure i que funciona com a aplicació Node. Wetty manté un socket



HTTP obert entre el client i el servidor, per obtenir uns temps de resposta ràpids, i per tant, es necessita disposar d'un programa al Servidor per realitzar la comunicació a través d'aquest Socket web.

Per això utilitzarem Node, que és un software que ens permet executar JavaScript al costat del servidor (Tradicionalment JavaScript és un llenguatge per al client). Node és molt popular en la programació de WebSockets i existeix gran quantitat de recursos, això fa que sigui la millor opció per a aquesta tasca.

### 6.3.7 NGINX i uWSGI



Figura 6.4: Logotip NGINX

És un servidor web i pròxy invers lleuger i de software lliure. L'usarem per a servir la nostra pàgina web quan estigui finalitzada. NGINX permet servir diferents peticions en paral·lel sense complicacions i té una configuració extremadament simple.

Més endavant parlarem al apartat de desenvolupament sobre com hem usat NGINX per servir la nostra pàgina web i a més fent-ho de forma segura, encriptant-ne el tràfic a la pàgina web a través d'HTTPS.



Figura 6.5: Logotip uWSGI

NGINX servirà l'aplicació usant el servidor d'aplicacions uWSGI, un petit software molt senzill de configurar[10] que s'encarregarà d'administrar l'aplicació, engegar-la, parar-la i crear totes les instàncies del servei necessàries. Per tant, no serà NGINX el que serveixi directament l'aplicació, NGINX s'encarregarà de rebre les peticions segures, i passar-les al servidor més especialitzat de uWSGI, que serà l'encarregat de mostrar i administrar l'aplicació. Necessitem usar un servidor d'aplicacions perquè NGINX és capaç de servir pàgines estàtiques, però una aplicació dinàmica com la nostra necessita usar un servidor especialitzat.



Figura 6.6: Logotip Let's Encrypt

### 6.3.8 Let's Encrypt i Certificats digitals

Com hem mencionat, per tal de garantir la seguretat a la nostra aplicació, utilitzarem HTTPS. Per tal de garantir al usuari aquesta seguretat, necessitarem uns certificats, que obtindrem a través de la Autoritat certificadora Let's Encrypt. Let's Encrypt és una Autoritat certificadora sense anim de lucre, fundada per la Electronic Frontier Foundation, Mozilla, la Universitat de Michigan i el Internet Security Research Group. El seu objectiu és aconseguir que no hi hagi tràfic web sense encriptar, i per aconseguir-ho, tenen eines per simplificar la obtenció de certificats, que a més són gratuïts, sempre garantint la màxima seguretat.

Actualment forma part de la Linux Foundation i es manté amb el suport de donacions de particular i empreses com Cisco, Akamai, Mozilla, Facebook, Digital Ocean i altres gegants del sector de les comunicacions.

Veurem amb més detall com obtenim els certificats i els afegim a la pàgina web al apartat de desenvolupament del projecte.

### 6.3.9 PostgreSQL



Figura 6.7: Logotip PostgreSQL

Com a base de dades del nostre projecte hem decidit utilitzar PostgreSQL, ja que és el sistema amb el que estem més familiaritzats, apart de ser un sistema lleuger i de software lliure.

Per administrar la base de dades, al tenir un esquema senzill (que veurem mes endavant en el apartat de desenvolupament), hem decidit simplement usar la interfície de psql a través de la terminal, ja que les operacions de manteniment no requereixen més que un parell de línies d'SQL.

### 6.3.10 SQLAlchemy



Figura 6.8: Logotip SQLAlchemy

Finalment, en la nostra aplicació, hem decidit usar SQLAlchemy per servir d'Interfície entre Python i la base de dades. Gràcies a SQLAlchemy podem accedir i consultar els elements de la base de dades com si es tractés d'objectes de Python, simplificant enormement el tractament de les dades emmagatzemades i ajudant-nos a evitar errors de programació i seguretat. Veurem exemples de com s'usa SQLAlchemy en veure amb més detall el funcionament de l'aplicació.

## 6.4 Desenvolupament

### 6.4.1 Disseny de l'estructura de la pàgina

La pàgina web que dissenyarem, serà utilitzada pels administradors del sistema com a eina per controlar el funcionament dels Nodes i per administrar-los si cal. A més, també és interessant que disposi d'una part pública on els voluntaris i el públic general puguin veure quins CAPTOR hi ha actualment en funcionament.

Hem dissenyat l'estructura de la figura 6.9 per a la nostra pàgina web:

Les pàgines principals que es crearan són les demostrades a continuació:

- **Pàgina principal (/):** Aquesta pàgina mostra les opcions per accedir a les diferents eines del sistema. Així com enllaços al sistema de registre i log-in. La opció per afegir un nou node només es mostra en el cas de que l'hagi accedit com a administrador.
- **Dashboard (/dashboard):** En aquesta pàgina es mostra una taula amb la informació rellevant de cada CAPTOR a primera vista. L'estat de la connexió del node amb el sistema, l'estat de connexió a la xarxa cel·lular... També disposa d'una barra d'eines per als administradors.
- **New CAPTOR (/alta):** Aquesta pàgina conté un formulari per entrar la informació necessària del node. Requereix permisos d'administrador.
- **Tools (/tools):** Pàgina sense ús determinat, per a eines de debug, tests de eines noves... Per provar i testejar elements nous del sistema en un entorn definitiu però sense modificar les vistes que usen els responsables dels nodes.

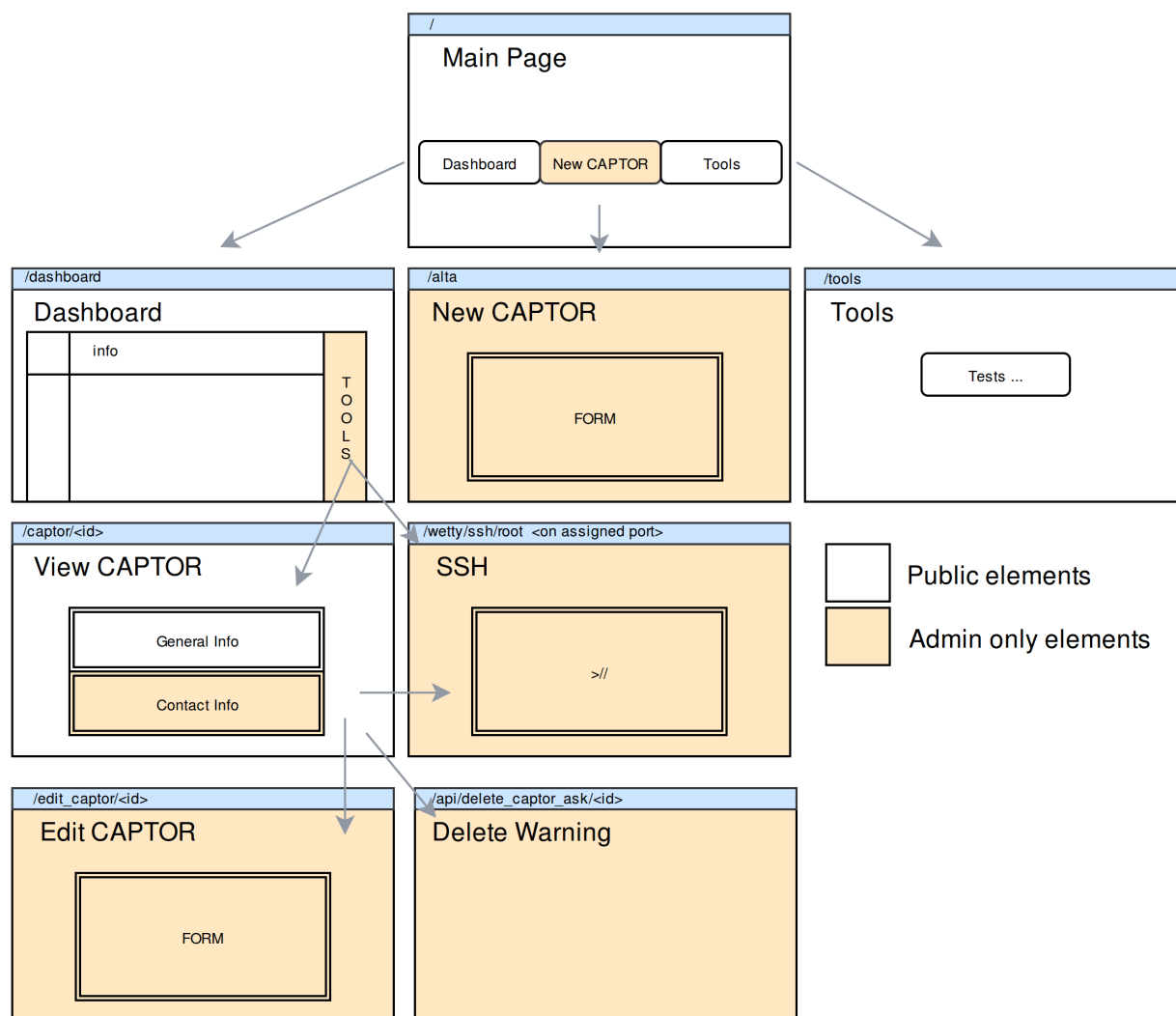


Figura 6.9: Estructura de la pàgina web, amb la URL i els permisos necessaris

- **View CAPTOR** (`/captor/<id>`): Aquesta pàgina mostra la informació general d'un node CAPTOR, la seva localització es mostra en un mapa, la informació de xarxa i l'estat de la connexió. Aquesta informació es publica a tothom. També té un últim apartat en el que es mostra la informació de contacte del voluntari encarregat del node. Aquesta informació, per respectar la privacitat del voluntari, és privada i tan-sols pot ser consultada per un administrador.
- **Edit CAPTOR** (`/edit_captor/<id>`): En aquesta pàgina web, es permet editar la majoria d'informació sobre un captor, des de la seva localització fins a la informació de xarxa. Requereix permisos d'administrador.
- **Delete Warning** (`/delete_captor_ask/<id>`): Eliminar un CAPTOR del sis-

tema és una acció definitiva, per tant, aquesta acció necessita ser acceptada i confirmada a través d'una pàgina en la que s'informa de les conseqüències. Requereix permisos d'administrador.

- **SSH (/wetty/ssh/root):** Aquesta pàgina es connectarà a través del port assignat al terminal (En detall en l'apartat del sistema SSH web) i mostrarà una consola semblant a un terminal Linux convencional amb comunicació directa al CAPTOR. Aquesta sessió és independent de la pàgina web i té una duració de 2 hores. Temps més que suficient per realitzar les tasques de manteniment típiques. Requereix permisos d'administrador.

### 6.4.2 Disseny de la base de dades

Com hem dit, la base de dades serà de tipus relacional, en PostgreSQL, aquesta és la seva estructura:

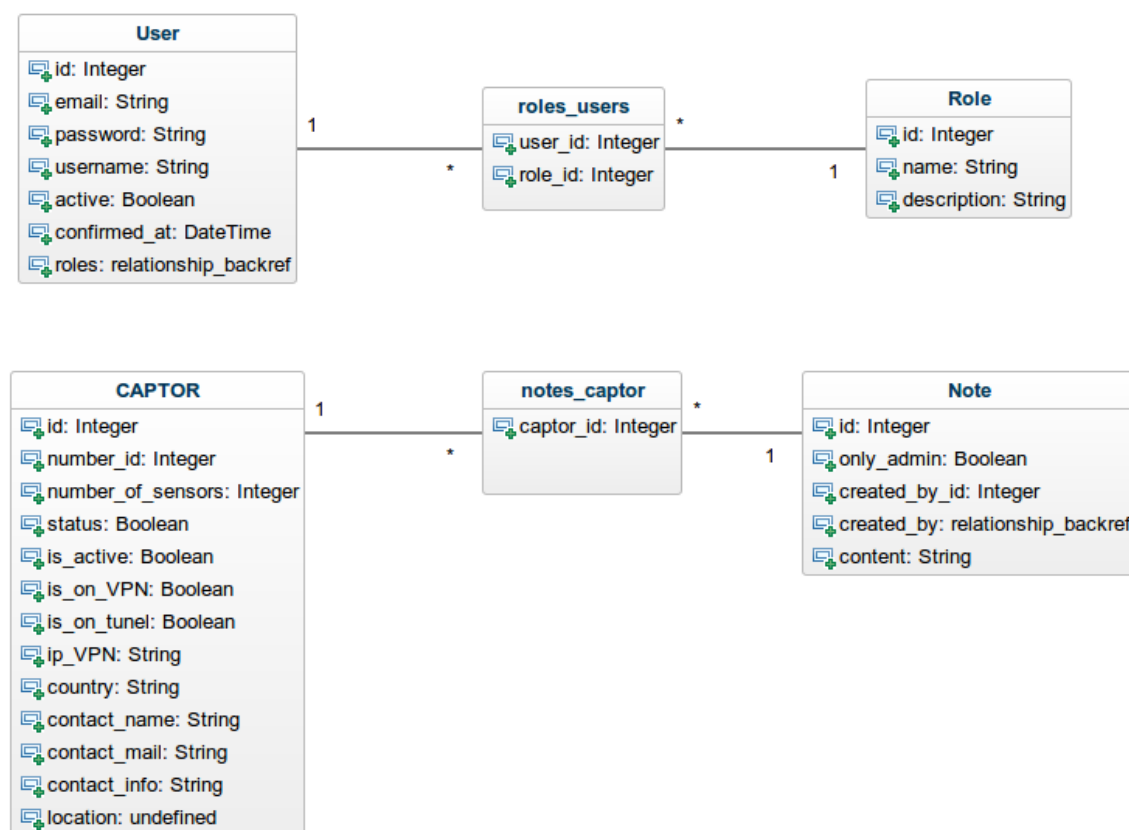


Figura 6.10: Estructura de la base de dades

Les classes `roles_users` i `notes_captor` són de tipus auxiliar per a crear les relacions entre `User` i `Roles` i `CAPTOR` i `Notes`. En la implementació final, la classe `Note` no forma part de la web ja que és enterament per a us intern d'scripts no relacionats i logs d'error, però com que la creem originalment dels de la pàgina web, l'afegim.

Per crear la base de dades, inicialment ens hem de connectar a la base elegida, n el nostre fitxer de configuració definim les següents variables, i definim la connexió amb SQLAlchemy:

---

```

1  #En el fitxer de configuració
2  SQLALCHEMY_DATABASE_URI = 'postgresql://postgres:contrassenya@localhost/captor'
3  SQLALCHEMY_TRACK_MODIFICATIONS = True
4
5  #En la nostra aplicació
6  db = SQLAlchemy(app)

```

---

Codi 5: Definim la connexió amb la base de dades

A continuació, podem procedir a definir les classes de python auxiliars que usarem per accedir al elements de la base de dades, i amb una instrucció, SQLAlchemy s'encarregarà de crear les taules necessàries a la base de dades per tal de que funcioni de la forma esperada:

---

```

1  #Taula roles_users
2  roles_users = db.Table('roles_users',
3      db.Column('user_id', db.Integer(), db.ForeignKey('user.id')),
4      db.Column('role_id', db.Integer(), db.ForeignKey('role.id')))
5
6  #Taula Role
7  class Role(db.Model, RoleMixin):
8      id = db.Column(db.Integer(), primary_key=True)
9      name = db.Column(db.String(80), unique=True)
10     description = db.Column(db.String(255))
11
12  #Taula User
13  class User(db.Model, UserMixin):
14      id = db.Column(db.Integer, primary_key=True)
15      email = db.Column(db.String(255), unique=True)
16      password = db.Column(db.String(255))
17      username = db.Column(db.String(255))
18      active = db.Column(db.Boolean())

```

```

19     confirmed_at = db.Column(db.DateTime())
20     roles = db.relationship('Role', secondary=roles_users, backref=db.backref('users',
    ↪     lazy='dynamic'))
21
22     #Taula Note
23     class Note(db.Model):
24         id = db.Column(db.Integer, primary_key=True)
25         is_only_me_and_admin = db.Column(db.Boolean())
26         created_by_id = db.Column(db.Integer, db.ForeignKey('user.id'))
27         created_by = db.relationship('User', backref=db.backref('notes', lazy='dynamic'))
28         captor_owner_id = db.Column(db.Integer, db.ForeignKey('captor.id'))
29         captor_owner = db.relationship('Captor', backref=db.backref('notes',
    ↪     lazy='dynamic'))
30         content = db.Column(db.String(500))
31
32     #Taula Captor
33     class Captor(db.Model):
34         id = db.Column(db.Integer, primary_key=True)
35         number_id = db.Column(db.Integer())
36         number_of_sensors = db.Column(db.Integer())
37         status = db.Column(db.Boolean())
38         is_active = db.Column(db.Boolean())
39         is_on_VPN = db.Column(db.Boolean())
40         is_on_tunel = db.Column(db.Boolean())
41         ip_VPN = db.Column(db.String(255))
42         country = db.Column(db.String(255))
43         contact_name = db.Column(db.String(255))
44         contact_mail = db.Column(db.String(255))
45         contact_info = db.Column(db.String(255))
46         location = db.Column(db.String(255))
47
48     #Creadora per a Usuaris
49     def __init__(self, c_number_id, c_number_of_sensors, c_status, c_is_active,
    ↪     c_is_on_VPN, c_is_on_tunel, c_ip_VPN, c_country, c_contact_name,
    ↪     c_contact_mail, c_contact_info, c_location):
50         self.number_id = c_number_id
51         self.number_of_sensors = c_number_of_sensors
52         self.status = c_status
53         self.is_active = c_is_active
54         self.is_on_VPN = c_is_on_VPN
55         self.is_on_tunel = c_is_on_tunel
56         self.ip_VPN = c_ip_VPN
    ↪     #IP VPN s ha canviat a ID en Emnify, Tenir en compte! (Endpoint ID)

```

```
57         self.country = c_country
58         self.contact_name = c_contact_name
59         self.contact_mail = c_contact_mail
60         self.contact_info = c_contact_info
61         self.location = c_location
62
63     #SQLAlchemy genera les instruccions i crea la taula
64     db.create_all()
```

---

Codi 6: Creem les classes necessàries per a la base de dades

SQLAlchemy també ens facilita enormement accedir als valors de la base de dades, a continuació un parell d'exemples:

---

```
1     #Crear i inserir un usuari
2     def create_user():
3         db.create_all()
4         user_datastore.create_user(email='admin@monitor.captor', password =
5             ↪ 'contrasenya')
6         db.session.commit()
7
8     #crear un rol i assignat-lo a un usuari
9     def create_role():
10         user_datastore.create_role(name='admin')
11         user_datastore.add_role_to_user('admin@monitor.captor', 'admin')
12         db.session.commit()
13
14     #fer una petició de tots els CAPTOR ordenats per id
15     all_captors = Captor.query.order_by(Captor.number_id).all()
```

---

Codi 7: Afegim elements i consultam la base de dades

### 6.4.3 Posada a punt de l'entorn

La pàgina web estarà allotjada a una màquina virtual en el CPD del departament d'Arquitectura de Computadors. Aquesta màquina executa una distribució Linux recomanada pel departament, Ubuntu LTS amb kernel 4.4.



```
monitor/
├── app/
│   ├── static/
│   │   └── images/
│   ├── templates/
│   │   ├── layouts/
│   │   └── security/
│   ├── __init__.py
│   └── views.py
├── flask/
├── wetty/
├── config.py
└── run_external.py
```

Figura 6.11: Estructura del directori monitor

Després d'instal·lar pip, Python i Flask segons les instruccions recomanades (En el nostre cas des dels repositoris oficials per defecte), crearem un directori monitor a */var*.

L'estructura d'aquest directori serà la de la Figura 6.11.

- **app/** : Conté els fitxers necessaris i que hem programat per a la nostra aplicació:
  - **static/** : Si a la nostra aplicació ens referim a alguna url que no detectem com una pàgina, s'assumirà que es un recurs estàtic (per exemple una imatge) i es buscarà en aquest directori. Per tant, aquí eés on posarem les imatges que volguem servir.
  - **templates/** : Coté tots els fitxers HTML de la nostra aplicació, al subdirectori *security* trobem els específics dels formularis de log-in, registre... I al directori *layouts* els generals de la nostra aplicació que usarem com a plantilla per a la resta de templates html:

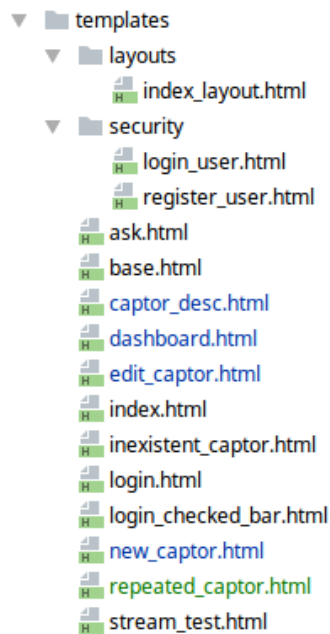


Figura 6.12: Contingut del directori templates

- **\_\_init\_\_.py** : Aquest fitxer especial de python, permet tractar el directori app com un sol executable. a la que s'intenti executar app (en el nostre cas des de *run\_external.py*), aquest serà el primer fitxer que es cridarà. Des d'aquí llancem la nostra aplicació.
  - **views.py** : Aquest fitxer conté tota la lògica Server-side del nostre servei. Programat en python Flask.
- **flask/** : Codi font de Flask.
  - **wetty/** : Codi font i configuracions de wetty.
  - **config.py/** : Document de configuració en el que es defineixen les variables que configuren la seguretat (tipus d'encryptació, contrassenyes), la base de dades (URL, credencials) i altres elements de la pàgina.
  - **run\_external.py/** : Aquest petit script s'encarrega d'importar l'aplicació, assegurar-se de que no hi hagi encara cap terminal corrent de l'últim cop que s'ha iniciat, i engega l'aplicació amb permisos d'accés des de l'exterior. Per tant, si executem *./run\_external* des del directori monitor, la nostra aplicació s'engegarà.

---

```
1  #!/flask/bin/python
2  from app import app
3  from subprocess import call
4
5  call(["killall", "timeout"]) #kill any shell that may be in use
6  app.run(host='0.0.0.0', threaded=True)
```

---

Codi 8: run\_external.py

#### 6.4.4 Flask: Servint les pàgines

Servir una pàgina en Flask és molt senzill, veiem l'exemple del codi 9.

---

```
1  @app.route('/captor/<id>')
2  @login_required
3  def captor_page(id):
4      currentUser = User.query.filter_by(id=current_user.get_id()).first()
5      captor = Captor.query.filter_by(number_id=id).first()
6      if(captor is None):
7          return render_template("inexistent_captor.html", auth=True, c_user=currentUser)
8      else:
9          lat = captor.location.split(',')[0]
10         lon = captor.location.split(',')[1]
11         print "lat: " + lat
12         print "lon: " + lon
13         if "admin" in currentUser.roles:
14             return render_template("captor_desc.html", auth=True, c_user=currentUser,
15                                     ↪ captor=captor, admin = True, lat = lat, lon = lon)
16         else:
17             return render_template("captor_desc.html", auth=True, c_user=currentUser,
18                                     ↪ captor=captor, admin = False, lat = lat, lon = lon)
```

---

Codi 9: Exemple de servir una pàgina en Flask

En la línia 1 del codi, veiem un decorador de python (`@app.route('/captor/<id>')`). Això ens indica que al accedir a `/captor/<id>` on `id` és la id d'un captor, cridarem aquesta funció i retornarem el resultat. Per tant, podem passar fàcilment paràmetres a la funció

directament des de la URL, i aquesta funció serveix de "plantilla" per totes les peticions amb el mateix format. Tornarem més endavant a aquest codi per veure el seu funcionament en detall.

Veiem que al retornar la funció cridem a la funció *render\_template()*. El que estem fent, és cridar a Jinja2 indicant que es generi una pàgina HTML a partir de la plantilla *captor\_desc.html* i passant uns paràmetres per usar en el renderitzat-ge de la plantilla. Un fragment d'aquesta plantilla en concret és així:

---

```

1  {% extends "login_checked_bar.html" %}
2  {% block content %}
3
4  <script>
5  $(document).ready(function(){
6      $('[data-toggle="tooltip"]').tooltip();
7  });
8  </script>
9
10 <script type="text/javascript">
11     var source = new EventSource('/streams/captor/{{captor.number_id}}');
12     source.onmessage = function(e){
13         [...]
14     };
15     function myMap() {
16         [...]
17     }
18 </script>
19 [...]
20 <div class="container-fluid">
21     <div class="row">
22         <div class="col-md-8">
23             <div class="row">
24                 <div class="col-md-6">
25                     <h1>
26                         CAPTOR {{captor.number_id}}
27                     </h1>
28                 </div>
29                 <div class="col-md-6">
30                     [...]
31                 </div>
32             </div>
33             <div class="row">
34                 <div class="col-md-6">
```

```

35         <div class="jumbotron well">
36             <h2>
37                 General Information
38             </h2>
39             <b>Number of sensors: </b> {{captor.number_of_sensors}}<br>
40             <b>Country:</b> {% if captor.country == "spain" %}Spain{% endif
41                 ↪ %}
42             {% if captor.country == "italy" %}Itally{% endif %}
43             {% if captor.country == "austria" %}Austria{% endif %}
44             {% if captor.country == "lab" %}Spain (UPC Lab){% endif %}
45             {% if captor.country == "palau" %}Spain (Palau Reial){% endif
46                 ↪ %}<br><br><br>
47             </div>
48         </div>
49         <div class="col-md-6">
50             <div class="jumbotron well">
51                 <h2>
52                     Network Information
53                 </h2>
54                 {% if captor.is_on_tunnel %}
55                 <b>Reverse SSH Tunnel:</b> Yes<br>
56                 <b>On port:</b> {{captor.number_id}}<br>
57                 {% else %}
58                 <b>Reverse SSH Tunnel:</b> No<br><br>
59                 {% endif %}
60                 {% if captor.is_on_VPN %}
61                 <b>Emnify VPN:</b> Yes<br>
62                 <b>Endpoint ID:</b> {{captor.ip_VPN}}
63                 {% else %}
64                 <b>Emnify VPN:</b> No<br><br>
65                 {% endif %}
66             </div>
67         </div>
68     </div>
69     <div class="col-md-4" id="map" style="height: 285px">
70         Mapa
71     </div>
72     {% if admin %}
73         [...]
74     {% endif %}
75 </div>

```

---

76 `{% endblock %}`

---

#### Codi 10: Fragment HTML de vista de CAPTOR

Com podem veure, els paràmetres que passavem al cridar *render\_template()* poden ser usats de forma molt simple aquí. Si es vol afegir un valor a la plantilla, només necessitem encapsular-lo en un parell de `{{ }}`. Un exemple es a la línia 53, en la que creem el títol de la pàgina amb el ID del captor: `CAPTOR {{captor.number_id}}`.

Les variables que hem passat però, també poden ser usades com a variables mateix. Per exemple, el codi entre les línies que reproduïm a continuació, comprova el valor *captor.is\_on\_tunel* i mostra un HTML diferent segons el valor:

---

```
1 {% if captor.is_on_tunel %}
2     <b>Reverse SSH Tunnel:</b> Yes<br>
3     <b>On port:</b> {{captor.number_id}}<br>
4 {% else %}
5     <b>Reverse SSH Tunnel:</b> No<br><br>
6 {% endif %}
```

---

#### Codi 11: Exemple de If/Else

### 6.4.5 Usuaris i Administradors

Com hem dit, la pàgina és accessible al públic en general, però reservem l'ús de certes funcionalitats als Administradors. Per tal de d'aconseguir això, com ja hem vist, els usuaris tindran un o diversos rols a la base de dades. Un usuari acabat de registrar, per defecte no disposarà de cap rol en concret.

Com hem vist al fragment de codi 7, podem afegir un rol a un usuari amb la funció *add\_role\_to\_user()*. Si un usuari té el rol *'admin'* serà considerat administrador i podrà veure les seccions reservades per als administradors.

Disposem de dos tipus de limitació diferent, cada una d'elles es controlarà d'una forma diferent:

- **Limitació de pàgina:** Una pàgina limitada no serà accessible per un usuari que no hagi accedit com a administrador al sistema. Si un usuari no-administrador intenta entrar a una pàgina restringida rebrà un error de seguretat i serà redirigit a la pàgina inicial.

Hem muntat el sistema per tal de que limitar l'accés a una pàgina restringida a sols usuaris administradors sigui molt senzill i no requereixi comprovacions dins de la pròpia lògica de la pàgina. Un exemple de pàgina limitada al administrador és la que ens permet afegir un captor nou:

---

```
1 @app.route('/alta')
2 @roles_required('admin')
3 def alta_page():
4     currentUser =
5         ↪ User.query.filter_by(id=current_user.get_id()).first()
6     return render_template("new_captor.html", auth=True,
7         ↪ c_user=currentUser)
```

---

Codi 12: Servint la pàgina d'afegir CAPTORS amb restricció a administradors

Com podem veure a la línia 2, afegim un decorator `@roles_required('admin')` a la funció. D'aquesta forma ens encarreguem de comprovar que l'usuari disposi del rol *'admin'* abans d'executar la funció.

- **Limitació de secció:** Una pàgina pot ser accessible a tots els usuaris en general, però pot tenir una secció que estigui limitada a administradors. Aquesta limitació la implementarem en Jinja2 directament, per tant, si un usuari no té el rol necessari, no s'inclourà la secció protegida en l'HTML que s'envia com a resposta. Un exemple en la pàgina que mostra la informació d'un captor:

---

```
1 currentUser = User.query.filter_by(id=current_user.get_id()).first()
2 if "admin" in currentUser.roles:
3     return render_template("captor_desc.html", auth=True,
4         ↪ c_user=currentUser, captor=captor, admin = True, lat =
5         ↪ lat, lon = lon)
6 else:
7     return render_template("captor_desc.html", auth=True,
8         ↪ c_user=currentUser, captor=captor, admin = False, lat =
9         ↪ lat, lon = lon)
```

---

Codi 13: Informat a Jinja2 que afegeixi el contingut d'administrador

Primer es comprova si l'usuari actual disposa del rol *'admin'*, i s'inclouen uns paràmetres o uns altres en la crida a jinja2. A continuació, en la secció d'html de la plantilla passada a Jinja2, comprovem si aquella part del HTML s'ha d'incloure o no:

---

```
1  {% if admin %}
2      <div class="row">
3          <div class="col-md-8">
4              <div class="jumbotron well">
5                  <h2>
6                      Contact Information
7                  </h2>
8                  <b>Contact:</b> {{captor.contact_name}}<br>
9                  <b>e-mail:</b> {{captor.contact_mail}}<br>
10                 <b>Other Contact Information:</b><br>
11                 ↪ <pre>{{captor.contact_info}}</pre><br>
12                 </div>
13             </div>
14         </div>
15     {% endif %}
```

---

Codi 14: Fragment d'HTML que només s'inclou si s'és administrador

El resultat es que un usuari amb permisos d'administració veurà unes seccions de la pàgina que un usuari no administrador no podrà veure:



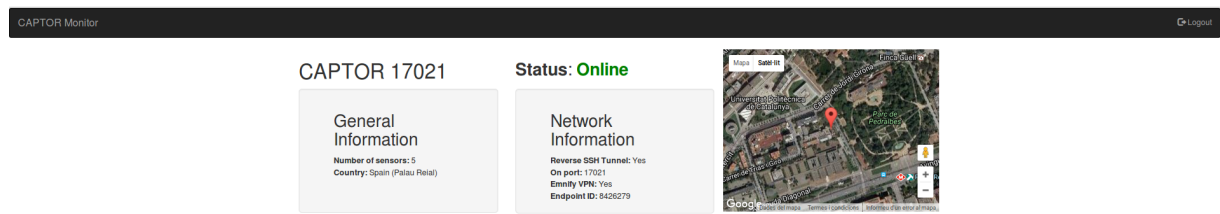


Figura 6.13: Vista de la pàgina d'informació per un usuari regular

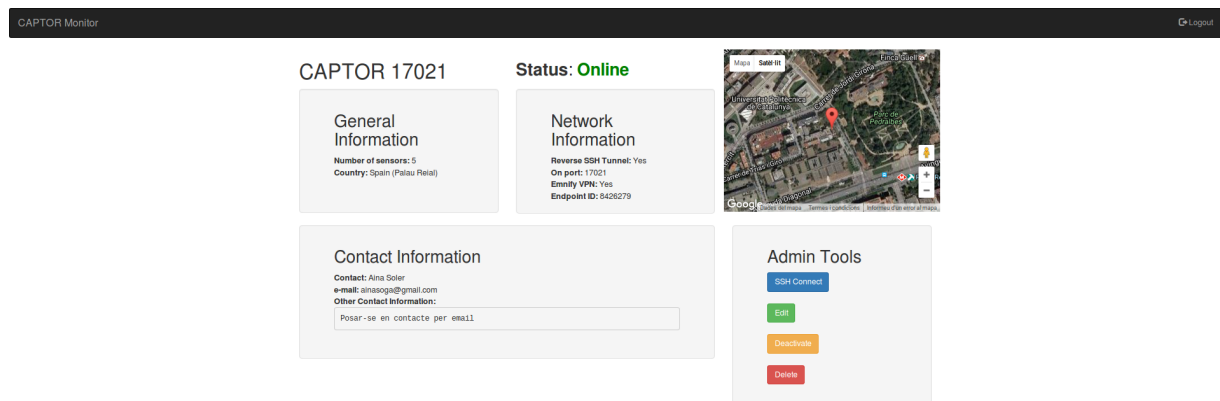


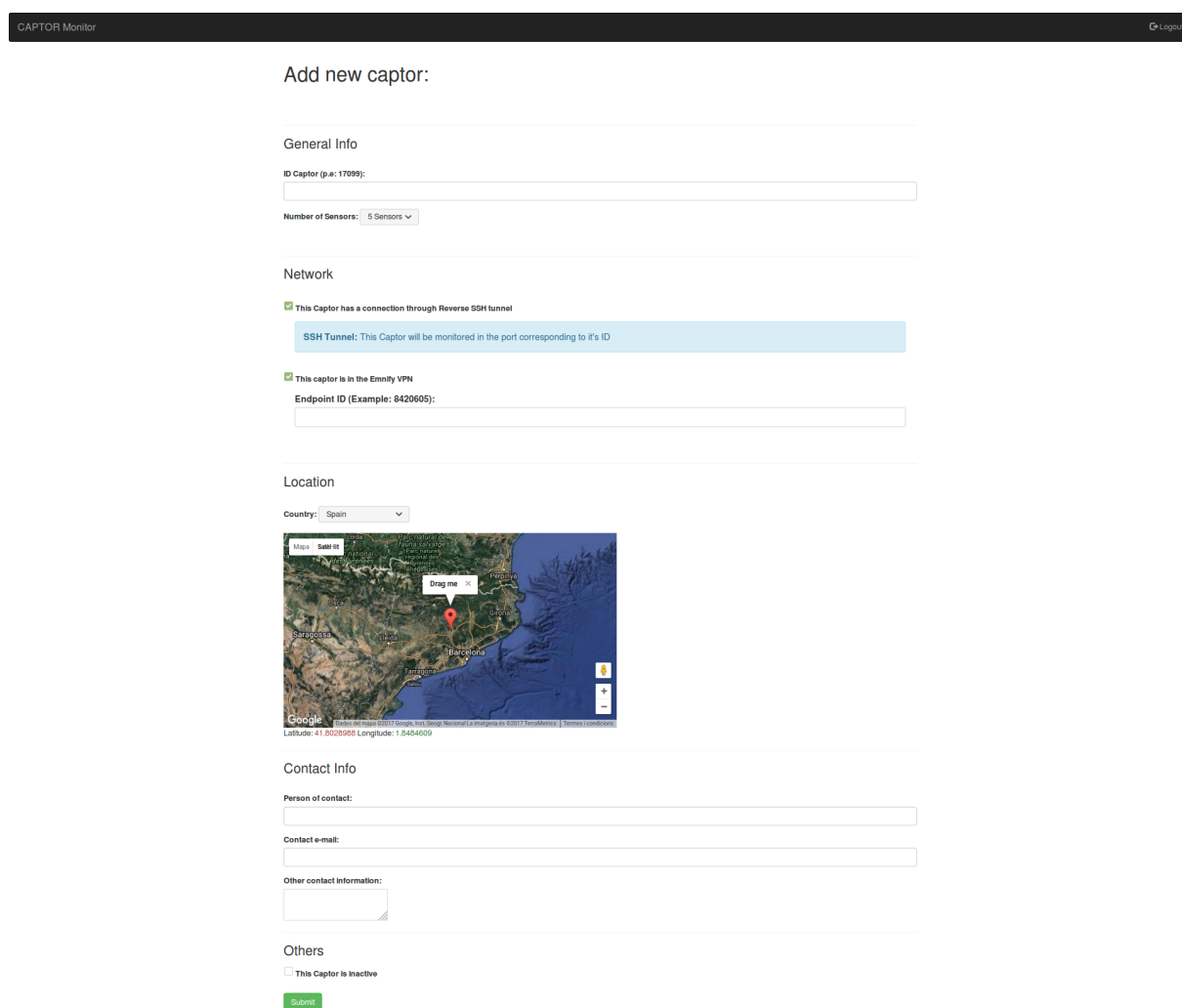
Figura 6.14: Vista de la pàgina d'informació per un usuari administrador

## 6.4.6 Afegir els CAPTORs

Cada cop que s'afegeix un CAPTOR al sistema, s'omple amb una informació que podrà ser consultada i editada en el futur pels administradors i el public general.

### 6.4.6.1 Formulari d'alta

Per inserir un CAPTOR nou al sistema, es presenta a l'usuari un formulari HTML que demana la informació necessària per tal de poder administrar el node. Aquesta pàgina es pot veure a la figura 6.15.



The screenshot displays the 'CAPTOR Monitor' web application. At the top, there is a dark header bar with the text 'CAPTOR Monitor' on the left and a 'Logout' link on the right. Below the header, the main content area is titled 'Add new captor:'. The form is organized into several sections: 'General Info' with fields for 'ID Captor (p.e: 17099):' and a 'Number of Sensors:' dropdown set to '5 Sensors'; 'Network' with checkboxes for 'This Captor has a connection through Reverse SSH tunnel' and 'This captor is in the Emnify VPN', and an 'Endpoint ID (Example: 8420605):' field; 'Location' with a 'Country:' dropdown set to 'Spain' and a Google Map of Barcelona with a red pin and 'Drag me' text; 'Contact Info' with fields for 'Person of contact:', 'Contact e-mail:', and 'Other contact information:'; and 'Others' with a checkbox 'This Captor is inactive'. A green 'Submit' button is at the bottom.

Figura 6.15: Vista del formulari per afegir un node

Els camps que es demanen són els següents:

- **ID Captor.**
- **Nombre de Sensors.**
- **El Node es troba connectat a Monitor per tunnel SSH?:** En cas de seleccionar aquesta opció, s'activarà l'accés SSH al node a través del port corresponent a la ID del node. Per tal de ser més intuïtiu, s'ha programat un estil CSS a aquest element utilitzant les noves animacions per tal de fer més entenedor i intuïtiu el fet de que és el activar i desactivar l'opció el que provoca que aparegui el quadre d'informació:

---

```
1 .reveal-if-active {
2   opacity: 0;
3   max-height: 0;
4   overflow: hidden;
5   font-size: 16px;
6   -webkit-transform: scale(0.8);
7     transform: scale(0.8);
8   -webkit-transition: 0.5s;
9     transition: 0.5s;
10 }
11 .reveal-if-active label {
12   display: block;
13   margin: 0 0 3px 0;
14 }
15 .reveal-if-active input[type=text] {
16   width: 100%;
17 }
18 input[type="radio"]:checked ~ .reveal-if-active, input[type="checkbox"]:checked ~
    ↪ .reveal-if-active {
19   opacity: 1;
20   max-height: 100px;
21   padding: 10px 20px;
22   -webkit-transform: scale(1);
23     transform: scale(1);
24   overflow: visible;
25 }
```

---

Codi 15: Classes CSS del desplegable

- **El Node està funcionant per 3G amb una SIM de la companyia Emnify?:** En cas de seleccionar aquesta opció, es desplegarà, de la mateixa manera que en el

camp anterior, un requadre que permet entrar la ID de la SIM del node (Proporcionada per la companyia telefònica).

- **País i Localització:** Un desplegable permet elegir entre: Espanya, Itàlia, Àustria, Espanya(Palau Reial) i Espanya (Laboratori). Al seleccionar una opció, el punter del mapa es centra de forma automàtica a una posició propera i permet ser arrossegat per tal de posicionar-lo a la localització definitiva.
- **Informació de contacte:** Persona i forma de contacte, també es disposa d'un quadre per afegir informació extra sobre el node.
- **Estat del Captor:** Ja que es permeten desar Nodes inactius al sistema, el CAPTOR es pot inserir com a inactiu.

Tota aquesta informació s'envia al servidor en un mètode POST, que recollim en una funció i inserim a la base de dades:

---

```

1  @app.route('/api/add_captor', methods =['POST'])
2  def add_captor():
3      c_number_id = request.form['number_id']
4      c_number_of_sensor = request.form['number_of_sensors']
5      [...]
6      create_and_insert_captor(c_number_id, c_number_of_sensor, c_status, c_is_active,
7                               ↪ c_is_on_VPN, c_is_on_tunel,
8                               c_ip_VPN, c_country, c_contact_name, c_contact_mail,
9                               ↪ c_contact_info, c_location)
10
11     return redirect('/', code=302)

```

---

Codi 16: Recollim per post i inserim a la base de dades. Fragment

#### 6.4.6.2 API de Google maps

Per tal d'aconseguir usar el mapa, usem la API de mapes de Google. Aconsegum una clau de la API a través d'un formulari de sol·licitud a la pàgina de APIs de Google, que podem veure a la figura 6.16.

A continuació, en la nostra pàgina, ens identifiquem a través d'una crida assíncrona a Google, que ens permetrà usar la seva API en aquesta pàgina a través de *myMap*:

---

```

1  <script async defer src="https://maps.googleapis.com/maps/api/js?key=[APIkey]=myMap"
2  type="text/javascript"></script>

```

---

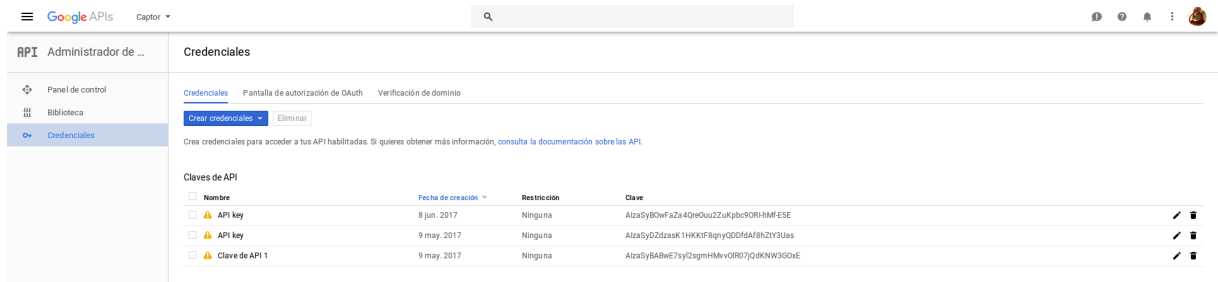


Figura 6.16: API de Google

## Codi 17: Identificant-nos a la API de Google Maps

Finalment, creem un mapa, i hi afegim un marcador. També hem desenvolupat diverses funcions auxiliars per fer la navegació en el mapa més fàcil.

```

1 function myMap() {
2
3     //creem un mapa, per defecte centrat a la primera opció (Espanya)
4     var latLng = new google.maps.LatLng(41.8028988, 1.8484609);
5     var map = new google.maps.Map(document.getElementById("map"), {
6         center: latLng,
7         zoom: 4,
8         mapTypeId: google.maps.MapTypeId.HYBRID
9     });
10
11     //creem un marcador en el centre del mapa
12     var marker = new google.maps.Marker({
13         position: latLng,
14         map: map,
15         draggable: true
16     });
17
18     [...]
19
20     marker.addListener('position_changed', function() { //Es mou el marcador
21         var lat = marker.getPosition().lat();
22         var lng = marker.getPosition().lng();
23         displayLatLng(lat, lng)
24     });
25
26     marker.addListener('dragend', function() { //S'arrossega el marcador

```

```
27     var lat = marker.getPosition().lat();
28     var lng = marker.getPosition().lng();
29 });
30
31 displayLatLng = function(lat, lng) { //Mostrem la posició als peus del mapa
32     var lat = parseFloat(lat).toFixed(7);
33     var lng = parseFloat(lng).toFixed(7);
34     $('#latitude').html(lat);
35     $('#longitude').html(lng);
36     document.getElementById("location").value = lat + ',' + lng;
37 }
38
39 panMapTo = function(lat, lng) { //Movem el mapa a una posició nova
40     var center = new google.maps.LatLng(lat, lng);
41     map.panTo(center);
42     marker.setPosition(center);
43 }
44
45 panMapToAndZoom = function(lat, lng, zoom) { //Movem el mapa + zoom
46     var center = new google.maps.LatLng(lat, lng);
47     map.setZoom(zoom);
48     map.panTo(center);
49     marker.setPosition(center);
50 }
51
52 //centrar el mapa cada cop que es seleccioni una nova localització
53 function foo(select) {
54     if(select.options[select.selectedIndex].getAttribute("value") == "spain"){
55         panMapToAndZoom(41.8028988, 1.8484609, 7);
56     }
57     [...] //Un per a cada opció
58 }
```

---

Codi 18: Creant un Mapa, un marcador i utilitats del mapa

#### 6.4.6.3 Consultar les dades del CAPTOR

Com ja hem vist en exemples anteriors, com a la figura 6.14, un usuari pot consultar la pàgina d'informació d'un captor. Aquesta informació s'obté de la base de dades a través d'una consulta que busca la ID i es mostra, en cas de que el CAPTOR no existeixi, es mostra una pàgina d'error.

### 6.4.7 Editar i eliminar CAPTORs

Un usuari administrador pot editar la informació d'un captor. Al entrar en aquest pàgina, es mostra al usuari un formulari semblant al de donada d'alta però en el que tota la informació que s'ha entrat es troba en els camps. Podem veure aquesta pàgina a la figura 6.17. Totes les modificacions sobre el CAPTOR, finalment, es comproven i s'insereixen a la base de dades.

CAPTOR Monitor Logout

### Editing Captor 17000

**General Info**

ID Captor (p.e: 17000):  
17000

Number of Sensors: 5 Sensors

**Network**

☐ This Captor has a connection through Reverse SSH tunnel

☒ This captor is in the Emnify VPN

Endpoint ID (Example: 8420605):  
8420605

**Location**

Country: Spain (UPC Lab)

Map: Satellite

Drag me

Latitude: 41.3895072 Longitude: 2.1128952

**Contact Info**

Person of contact:  
Aina Soler

Contact e-mail:  
ainasoga@gmail.com

Other contact information:  
This is a Test CAPTOR.  
Does not work

**Others**

☒ This Captor is inactive

Submit

Figura 6.17: Vista del formulari per editar un node

A més, un usuari administrador també té la capacitat d'eliminar un node del sistema, ja que aquesta acció no és reversible, es demanarà confirmació de l'usuari a través d'una pàgina on s'explica que no es pot desfer. Finalment, el captor s'elimina totalment de la base de dades amb la funció del codi 19.

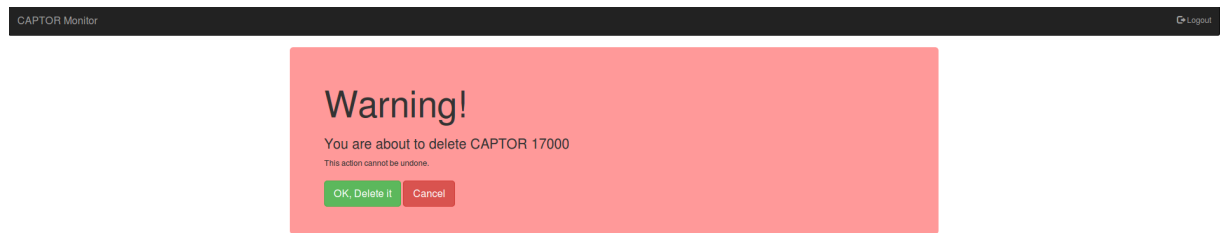


Figura 6.18: Avis al eliminar un node

---

```
1 @app.route('/api/delete/<id>', methods=['GET'])
2 def delete_captor(id):
3     del_captor = Captor.query.filter_by(number_id=id).first()
4     db.session.delete(del_captor)
5     db.session.commit()
6     return redirect('/', code=302)
```

---

Codi 19: Eliminant un CAPTOR de la base de dades

## 6.4.8 Dashboard

Aquesta pantalla mostra una llista de tots els nodes donats d'alta al sistema, així com el seu estat a la xarxa i informació de localització. Podem veure un exemple a la figura 6.19.

Per servir aquesta pàgina, accedim a la base de dades i retornem tots els nodes ordenats per ID. La resta d'informació (Estatut de les connexions) s'enviaran a través d'un Server-Sent Event, com veurem a continuació

### 6.4.8.1 Status del CAPTOR al sistema

Si un node esta funcionant correctament, el servidor del servei tindrà una connexió oberta en el port equivalent a la ID del node (Veurem aquest sistema en mes detall al pròxim Capítol) i per tant es podrà activar la funció de connexió directa amb el node.

Des d'una consola connectada al servidor, podem veure quins ports en el sistema estan oberts amb la comanda `netstat -ltun`. Aquesta comanda ens retorna un resultat com el que podem veure a la figura 6.20. En aquesta sortida hem de buscar tots els ports que es trobin entre 17000 i 17999 ja que aquest es el rang esperat de ID de CAPTORs. En l'exemple de la figura veiem connexió dels nodes 21, 22, 23, 24, 25, 26 i 27.



CAPTOR Monitor Logout

CAPTOR ID	Status	Location	3G Status	Tunnel Info	Tools
17000	Inactive	Spain (UPC Lab)	Inactive	Inactive	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17001	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17002	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17003	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17004	Offline	Spain (Palau Reial)	Offline	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17005	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17006	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17007	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17008	Offline	Spain (Palau Reial)	Attached	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17009	Online	Spain (Palau Reial)	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17010	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17011	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17012	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17013	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17014	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17015	Offline	Spain (Palau Reial)	Attached	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17016	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17017	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17018	Offline	Spain (Palau Reial)	Attached	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17019	Online	Spain (Palau Reial)	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17020	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17021	Online	Spain (Palau Reial)	Online	17021	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17022	No Info	Spain	Loading...	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17023	Online	Spain (Palau Reial)	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17024	Offline	Spain	Attached	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17025	Online	Spain (Palau Reial)	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17026	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>
17027	Online	Spain	Online	No Info	<a href="#">Complete Info</a> <a href="#">SSH Connect</a>

Figura 6.19: Dashboard amb nodes connectats del tot, amb problemes de 3G i desconnectats

```
admuser@monitor:/var/monitor$ netstat -ltun
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:5432          0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17021         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17022         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17023         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17024         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17025         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17026         0.0.0.0:*               LISTEN
tcp        0      0 127.0.0.1:17027         0.0.0.0:*               LISTEN
tcp6       0      0 :::22                   :::*                     LISTEN
tcp6       0      0 :::1:5432                :::*                     LISTEN
tcp6       0      0 :::1:17021                :::*                     LISTEN
tcp6       0      0 :::1:17022                :::*                     LISTEN
tcp6       0      0 :::1:17023                :::*                     LISTEN
tcp6       0      0 :::1:17024                :::*                     LISTEN
tcp6       0      0 :::1:17025                :::*                     LISTEN
tcp6       0      0 :::1:17026                :::*                     LISTEN
tcp6       0      0 :::1:17027                :::*                     LISTEN
udp        0      0 0.0.0.0:68              0.0.0.0:*
```

Figura 6.20: Estat dels ports al servidor

La manera més senzilla de trobar aquesta informació des del nostre codi, però, es consultar el fitxer `/proc/net/tcp6` que podem veure a la figura 6.21. Si mirem la columna `local_address`, veiem el port en hexadecimal.

```
admsuser@monitor:/var/monitor$ cat /proc/net/tcp6
```

sl	local_address	remote_address	st	tx	queue	rx	queue	tr	tm->when	retrnsmt	uid	timeout	inode
0:	00000000000000000000000000000000:0016	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	0	0	18446635 1 0000000000000000 100 0 0 10 0
1:	00000000000000000000000000000000:1538	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	111	0	1641730 1 0000000000000000 100 0 0 10 0
2:	00000000000000000000000000000000:427D	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27745521 1 0000000000000000 100 0 0 10 0
3:	00000000000000000000000000000000:427E	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27986540 1 0000000000000000 100 0 0 10 0
4:	00000000000000000000000000000000:427F	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27986662 1 0000000000000000 100 0 0 10 0
5:	00000000000000000000000000000000:4280	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27986813 1 0000000000000000 100 0 0 10 0
6:	00000000000000000000000000000000:4281	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27986828 1 0000000000000000 100 0 0 10 0
7:	00000000000000000000000000000000:4282	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27986184 1 0000000000000000 100 0 0 10 0
8:	00000000000000000000000000000000:4283	00000000000000000000000000000000:0000	0A	00000000	00000000	00	00000000	00000000	00000000	00000000	1001	0	27986380 1 0000000000000000 100 0 0 10 0

```
admsuser@monitor:/var/monitor$
```

Figura 6.21: Contingut de `/proc/net/tcp6`

Des de python, podem consultar aquesta informació utilitzant la llibreria *psutil* que son eines per consultar contingut de fitxers localitzats a `/proc/`. Per consultar i filtrar quins nodes estan connectats, ho fem de la següent manera:

---

```
1 def get_current_captors():
2     result = ""
3     for con in psutil.net_connections(kind="tcp"):
4         if con.laddr[1] >= 17000 and con.laddr[1] < 18000 and con.laddr[0]
5             == '127.0.0.1':
6                 result = result + str(con.laddr[1]) + " "
7     return result
```

---

Codi 20: Obtenint nodes connectats

Aquesta informació, però, s'enviarà al client en forma de Server-Sent Event. Com ja hem explicat, un Server-Sent Event és una comunicació passiva que s'estableix entre el client i el servidor i que permet al servidor enviar dades cap al client en qualsevol moment. Un resum d'aquesta comunicació es pot veure a la figura 6.22

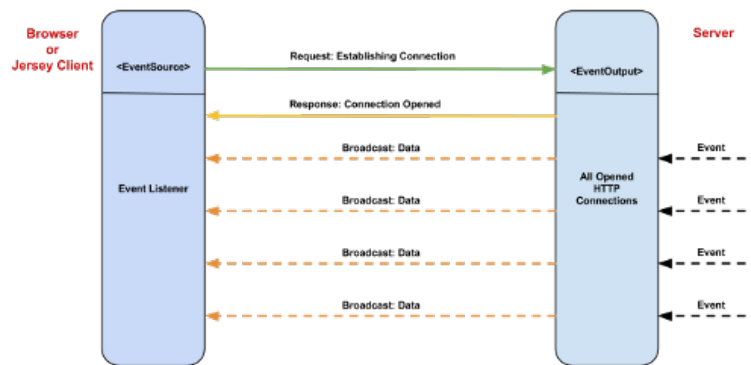


Figura 6.22: Missatges enviats en un SSE

Font: Sunil Gulabani - RESTful Web Services (Llibre)

El primer pas per establir aquesta comunicació, és, des del client, amb un script de JavaScript, demanar subscripció al event:

---

```

1  //Ens subscriuim al event
2  var source = new EventSource('/streams/all_captors');
3  //Al rebre un missatge en aquest event...
4  source.onmessage = function(e){
5      //Parsejem les dades
6      info = JSON.parse(e.data);
7      console.log(info)
8      {% for captor in all_captors %}
9          if({{captor.number_id}} in info){
10             if(info[{{captor.number_id}}] == "ONLINE"){
11                 [...] //Modifiquem l'estil de la pàgina per mostrar el node online
12             }
13             else if(info[{{captor.number_id}}] == "OFFLINE"){
14                 [...] //Modifiquem l'estil de la pàgina per mostrar el node offline
15             }
16             else {
17                 [...]
18                 ↪ //Encara no hem rebut informació del node correcta ("Loading...")
19             }
20             {% endfor %}
21         };

```

---

Codi 21: Petició de subscripció al event

Pel que fa al servidor, el primer pas que hem de fer, és contestar a la petició d'stream amb l'stream demanat, indicant en el mimetype que es tracta d'un stream d'events:

---

```

1 @app.route('/streams/all_captors')
2 def stream_all():
3     captor_list = Captor.query.order_by(Captor.number_id).all()
4     return Response(event_stream_all_captors(captor_list),
        ↳ mimetype="text/event-stream")

```

---

#### Codi 22: Resposta a la petició d'event stream

I finalment hem d'iniciar l'stream mateix. Un stream és un Loop infinit (Amb un delay, per no sobrecarregar el client ni el servidor). Que consultarà l'estat de cada captor, el posarà en una estructura amb el format correcte i l'enviarà. Per enviar la resposta sense retornar la funció, usarem la instrucció per defecte de Python *"yield"*:

---

```

1 def event_stream_all_captors(captor_list):
2     while True:
3         data_to_send = "data: {\n"
4         for captor in captor_list:
5             if is_online_dis_captor(captor.number_id):
6                 data_to_send = data_to_send + 'data: "' + str(captor.number_id) +
                    ↳ '" : "ONLINE",\n'
7             else:
8                 data_to_send = data_to_send + 'data: "' + str(captor.number_id) +
                    ↳ '" : "OFFLINE",\n'
9
10        data_to_send = data_to_send[:-2] + "\ndata: }\n\n"
11        yield data_to_send
12        time.sleep(2)

```

---

#### Codi 23: Server-Sent Event stream

Si observem aleshores l'estatus de la consola del navegador al visitar la pàgina, observarem que cada 2 segons, rebem un missatge amb l'estatus de cada node:

Object { 17000: "OFFLINE", 17001: "OFFLINE", 17002: "OFFLINE", 17003: "OFFLINE", 17004: "OFFLINE", 17005: "OFFLINE", 17006: "OFFLINE", 17007: "OFFLINE", 17008: "OFFLINE", 17009: "OFFLINE", 18 més... } dashboard:74:13

Figura 6.23: Missatges rebuts cada 2 segons

### 6.4.8.2 API d'Emnify: Status del CAPTOR a la xarxa cel·lular

L'últim camp de la taula que veurem en detall, és l'estatus del node a la xarxa 3G. L'ISP contractat per a proveir 3G als nodes (Emnify) disposa d'una API per consultar aquestes dades, que un cop obtingudes, enviarem al client en un Server-Sent Event similar al anterior. Per tant, en aquesta secció sols veurem en detall com obtenim les dades de la API segura d'Emnify.

Emnify utilitza un procés d'autenticació per accedir a aquestes dades. El procés es resumeix en el diagrama següent:

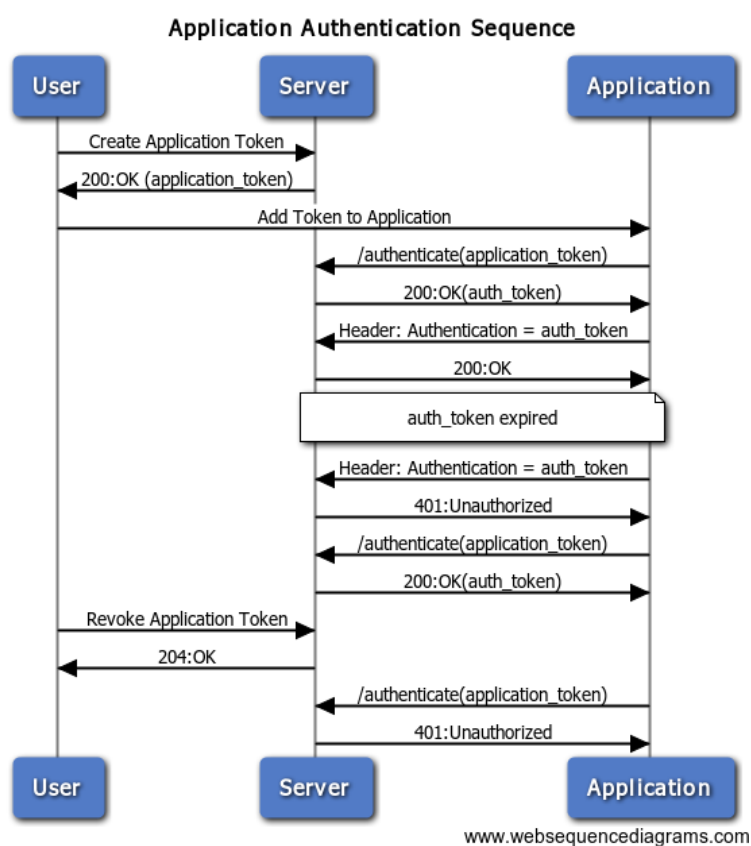


Figura 6.24: Sequència d'autorització d'aplicació a Emnify

És a dir, hem d'obtenir un Token d'aplicació que afegirem a l'aplicació. L'aplicació haurà d'usar aquest token per obtenir un token d'autorització amb una duració de 30 minuts. Aquest token, aleshores, podrà ser usat per l'aplicació per obtenir dades a través de la API.

A través de la pàgina web d'Emnify, sol·licitem un Token d'Aplicació al que donem permís d'accés a aquestes dades en concret.

Application Tokens					+ Create Application Token	
Show Revoked Application Tokens						
ID	Created	Created by	Purpose	Status		
376	2017-05-16T10:53:36+0000	Aina Soler	Request status of the connectivity of our endpoints	Activated	✖	

Figura 6.25: Sollicitud de Token d'aplicació

A continuació, afegim aquest token a la nostra aplicació, i l'usem per sol·licitar un token d'autorització:

---

```

1  #Token d'Aplicació obtingut a la web
2  API_APP_Token = "eyJhbGciOiJIc2E"
3
4  #Demanem un token d'autorització usant el token d'autenticació
5  def request_API_key():
6      res = requests.post('https://cdn.emnify.net/api/v1/authenticate',
7          ↪ headers={"Content-Type": "application/json"}, json = {"application_token":
8          ↪ API_APP_Token})
9      authkey = res.json()['auth_token']
10     return authkey

```

---

Codi 24: Obtenció d'un token d'autorització

Finalment, podem usar aquest token d'autorització per sol·licitar les dades desitjades de la API. Com que és un procés lent, aquesta informació només s'actualitzarà al client un cop per minut:

---

```

1  if captor.is_on_VPN:
2      res = requests.get('https://cdn.emnify.net/api/v1/endpoint/' + str(captor.ip_VPN)
3          ↪ + '/connectivity',
4          ↪ headers={"Authorization": "Bearer " + my_key})
5
6      print (res.json()["status"]["description"])
7      #S'imprimirà ONLINE, OFFLINE, o ATTACHED

```

---

Codi 25: Obtenint informació desitjada de la API

### 6.4.9 Connexió directa per SSH a través de la web

La última funció que veurem és la connexió al CAPTOR per SSH a través de la web. Com hem dit (i veurem en detall més endavant) un node que funciona correctament manté

una connexió establerta en un part del servidor corresponent a la seva ID. A través d'aquest port, podem establir una connexió SSH amb el node.

Per tal de realitzar aquesta funció, usarem Wetty, un emulador de terminal per a navegador. Aquest emulador utilitza NodeJS com a backend, i rep com a paràmetres en quin port iniciar-se i a on connectar-se.

Per tal de que la connexió sigui independent del servei web, hem decidit que cada connexió, obrirà una còpia del servei en un port lliure durant dues hores. Per dur a terme això, usarem la comanda estandar de unix *timeout*.

---

```
1 @app.route('/ssh/<id>')
2 @roles_required('admin')
3 def ssh_to(id):
4     if (is_online_dis_captor(id)):
5         #busquem un port lliure
6         port_to_open = int(id)
7         while (is_online_dis_captor(port_to_open)):
8             port_to_open = port_to_open + 1000
9         #iniciem el servei en el port lliure
10        process = Popen(['timeout', '2h', 'node', './wetty/app.js', '-p',
11                        ↪ str(port_to_open), '--sshport', str(id)])
12        #Redirigim a l'usuari a la tti
13        time.sleep(2) #temps suficient per encendre tots els serveis
14        return
15        ↪ redirect('http://monitor.pc.ac.upc.edu:'+str(port_to_open)+'wetty/ssh/root',
16        ↪ code=302)
17        return "ok"
18    else:
19        return "Error, captor not online"
```

---

#### Codi 26: Iniciant una sessió SSH

Analitzem el codi 26: Entre les línies 6 i 8, buscarem un port lliure (re-aprofitant les funcions de mirar si un captor es troba connectat) sumant 1000 consecutivament al port del captor. Per tant, si ens volem connectar al captor 17021, la primera connexió s'establirà amb una terminal accessible a través del port 18021. Si intentem establir una nova connexió abans de que hagi caducat la anterior, s'establirà al port 19021...

Un cop hem obtingut un port lliure, iniciem el servei amb un timeout de 2 hores, indicant usar el port lliure i que volem que la connexió sigui cap al port corresponent a la ID del node.

Finalment, redirigim la connexió al propi servidor però a través del port lliure obtingut,

indicant que volem que s'intenti accedir com a root. On es sol·licitarà la contrasenya al usuari i s'establirà una connexió amb el servidor.

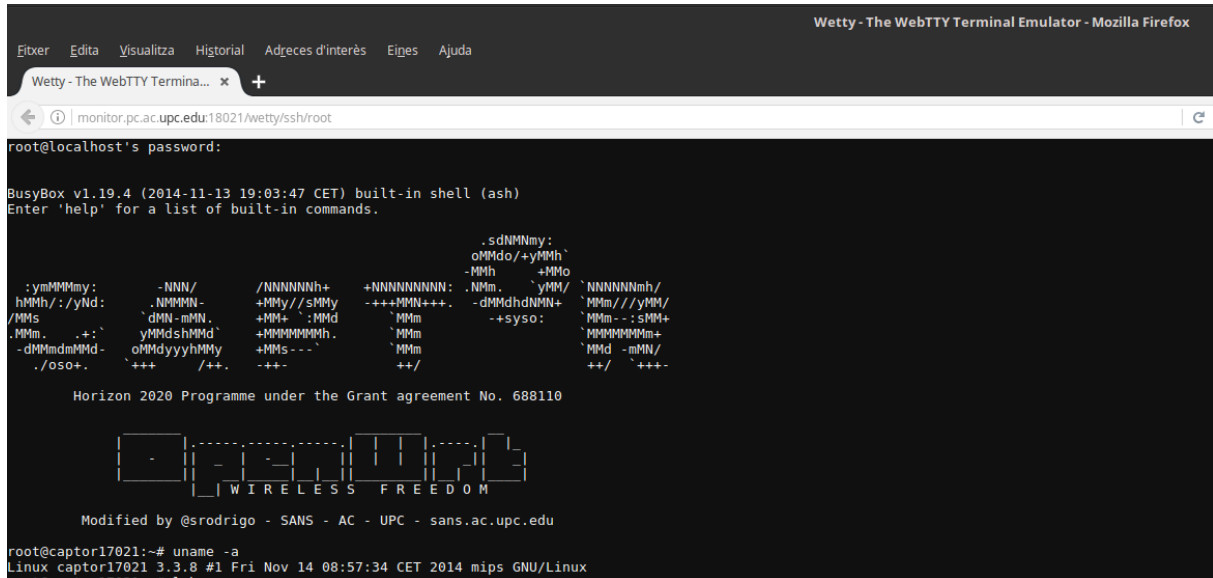


Figura 6.26: Sessió SSH establerta amb el node

## 6.4.10 Deployment i HTTPS

Finalment, veurem com servim la pàgina amb el nostre servidor d'aplicacions.

### 6.4.10.1 Servint l'aplicació amb uWSGI

En primer lloc, hem de crear un descriptor de punt d'entrada a la aplicació WSGI. WSGI és una interfície estàndard per a servidors de pàgines web en python[11], per tant, aquest mateix fitxer, ens serveix si mai decidim usar un servidor d'aplicacions diferent.

Aquest fitxer és el del codi 27.

```
1 from app import app as application
2
3 if __name__ == "__main__":
4     application.run()
```

Codi 27: WSGI Entry Point Descriptor

A nontinuació, creem un fitxer de configuració per al servidor d'aplicacions amb la configuració del codi 28



---

```
1 [uwsgi]
2 module = wsgi:application
3
4 master = true
5 processes = 5
6
7 socket = master.sock
8 chmod-socket = 660
9 vacuum = true
10
11 die-on-term = true
```

---

#### Codi 28: WSGI Config File

Els camps notables d'aquest fitxer de configuració són:

- **master i processes:** Iniciem el servidor en mode master i reservem 5 workers per servir les peticions.
- **socket:** Ja que les connexions seran rebudes per Nginx, que passarà les peticions al servidor d'aplicacions. Com que aquests components treballen al mateix servidor, usarem un socket d'Unix per les comunicacions entre ells, ja que és més segur i ràpid.
- **vacuum:** Amb aquesta opció, ens assegurem de que el procés neteja el socket al aturar-se.
- **die-on-term:** Ja que la nostra aplicació s'executarà en iniciar el servidor, amb aquesta opció ens assegurem que el sistema de init i el servidor de l'aplicació interpreten les senyals de la mateixa manera.

És interessant veure que no especifiquem al servidor d'aplicacions quin protocol està usant l'aplicació en concret. De forma nativa uWSGI usa un protocol ràpid binari anomenat *uwsgi*, Nginx també pot interpretar aquest protocol, així que es millor usar-ho d'aquesta manera que forçar la comunicació per HTTP.

##### 6.4.10.2 Iniciar l'aplicació des de systemd

Ja que desitjem que la pagina s'iniciï al engegar el servidor, creem un servei de systemd que s'encarregui d'engegar-la. Per tant, definim el servei al fitxer `"/etc/systemd/system/-monitor.service"`, els continguts del qual son en el codi 29

---

```
1 [Unit]
2 Description=instancia uWSGI per servir Monitor
3 After=network.target
4
5 [Service]
6 User=admuser
7 Group=www-data
8 WorkingDirectory=/var/monitor
9 Environment="PATH=/var/monitor/flask/bin"
10 ExecStart=/var/monitor/flask/bin/uwsgi --ini monitor.ini
11
12 [Install]
13 WantedBy=multi-user.target
```

---

Codi 29: Contingut de monitor.service

- **[Unit]:** En aquesta secció, incloem una descripció de servei, i el init target. Systemd no iniciarà el servei fins haver superat el network.target, és a dir, haver iniciat amb èxit la xarxa.
- **[Service]:** La informació sobre el grup al que pertany el procés (www-data, per facilitar la comunicació amb Nginx) i el procés a iniciar
- **[Install]:** Finalment, indiquem a systemd que volem que s'iniciï el procés quan s'hagi acabat de bootar en mode multi-user.

Finalment, activem el servei usant les comandes de systemd:

---

```
1 sudo systemctl start monitor
2 sudo systemctl enable monitor
```

---

Codi 30: Activant el servei

#### 6.4.10.3 Configurant NGINX per passar les peticions

En aquest moment, el nostre servidor uWSGI està esperant peticions al socket que hem indicat. Aquestes peticions les passarem a través d'NGINX.

Creem un fitxer de configuració en el directori d'NGINX */etc/nginx/sites-available/monitor*. En aquest bloc, indicarem que escoltem peticions al port 80 del nostre servidor i que les passarem totes a uWSGI a través del socket que hem indicat anteriorment.

```
1 server {  
2     listen 80;  
3     server_name monitor.pc.ac.upc.edu;  
4  
5     location / {  
6         include uwsgi_params;  
7         uwsgi_pass unix:/var/monitor/master.sock;  
8     }  
9 }
```

Codi 31: Configuració del block de servidor NGINX

Finalment, activem el bloc creant-ne un link al directori de llocs actius d'NGINX:  
`sudo ln -s /etc/nginx/sites-available/monitor /etc/nginx/sites-enabled.`

#### 6.4.10.4 Obtenint esl certificats

Com hem dit, assegurarem la connexió al nostre servei a través d'HTTPS. I per a obtenir els certificats, usarem el sistema simplificat de Let's Encrypt. El procés que seguirà la nostra pàgina per certificar la seva seguretat és resumit a la figura 6.27.

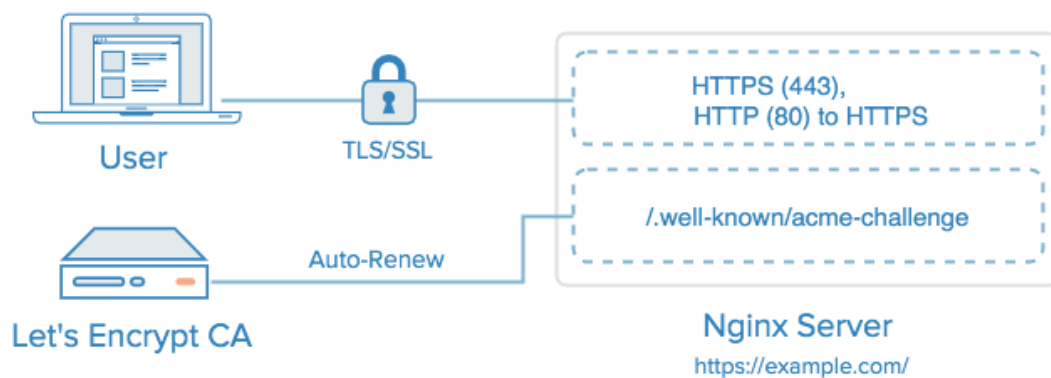


Figura 6.27: Funcionament d'un certificat de Let's Encrypt en el servidor Nginx

**Font:** Digital Ocean

Let's Encrypt, per certificar la nostra identitat, accedeix al directori `./well-known` del nostre servidor, on espera trobar-hi un contingut en concret.

Per tant, el primer que hem de fer és permetre l'accés a aquest directori afegint aquest bloc a NGINX:

---

```
1 location ~ /\.well-known {  
2     allow all;  
3 }
```

---

#### Codi 32: Configuració del block de servidor NGINX

A continuació, iniciem el procés de certificació amb la comanda de bash:

---

```
1 sudo certbot certonly --webroot --webroot-path=/var/www/html -d monitor.pc.ac.upc.edu
```

---

#### Codi 33: Iniciant el servei de certbot

Això accedirà al directori proporcionat amb un repte, que el directori haurà de respondre correctament. Aquest procés és automàtic.

### 6.4.10.5 Generant un Diffie-Hellman Group segur

Per millorar la nostra seguretat, generem un grup Diffie-Hellman de 2048 bits. Per fer-ho usem la comanda de bash `sudo openssl dhparam -out /etc/ssl/certs/dhparam.pem 2048`.

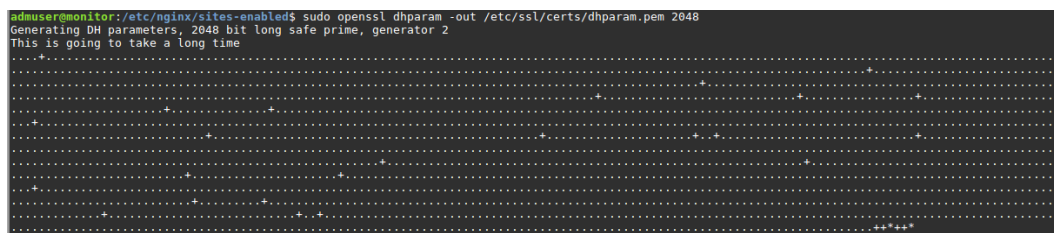


Figura 6.28: Generant un grup Diffie-Hellman de 2048 bits

### 6.4.10.6 Assegurant la nostra pàgina web amb TLS/SSL

Finalment, ajuntem els certificats amb les claus generades anteriorment per assegurar la nostra pàgina, per fer-ho, tansols hem de substituir el bloc del codi 32 per el que mostrem a continuació. El que fa aquest bloc és redirigir totes les peticions a HTTPS. I un cop redirigides, les assegura amb les claus i les firma amb els certificats.

---

```
1 server {
2     listen 80;
3     server_name monitor.pc.ac.upc.edu;
4     return 301 https://$server_name$request_uri;
5 }
6
7 server{
8     listen 443 ssl;
9     server_name monitor.pc.ac.upc.edu;
10
11     ssl_certificate /etc/letsencrypt/live/monitor.pc.ac.upc.edu/fullchain.pem;
12     ssl_certificate_key /etc/letsencrypt/live/monitor.pc.ac.upc.edu/privkey.pem;
13
14     ssl_protocols TLSv1 TLSv1.1 TLSv1.2;
15     ssl_prefer_server_ciphers on;
16     ssl_dhparam /etc/ssl/certs/dhparam.pem;
17     ssl_ciphers 'ECDHE-RSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-
18     RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:DHE-RSA-AES128-[...]'
19     ssl_session_timeout 1d;
20     ssl_session_cache shared:SSL:50m;
21     ssl_stapling on;
22     ssl_stapling_verify on;
23     add_header Strict-Transport-Security max-age=15768000;
24
25     location / {
26         include uwsgi_params;
27         uwsgi_pass unix:/var/monitor/monitor.sock;
28     }
29 }
```

---

Codi 34: Configuració del block de servidor NGINX, amb TLS/SSL

Si accedim a la nostra pàgina, podem veure que el navegador mostra el candau verd que ens informa de que la pàgina està correctament encriptada i firmada.

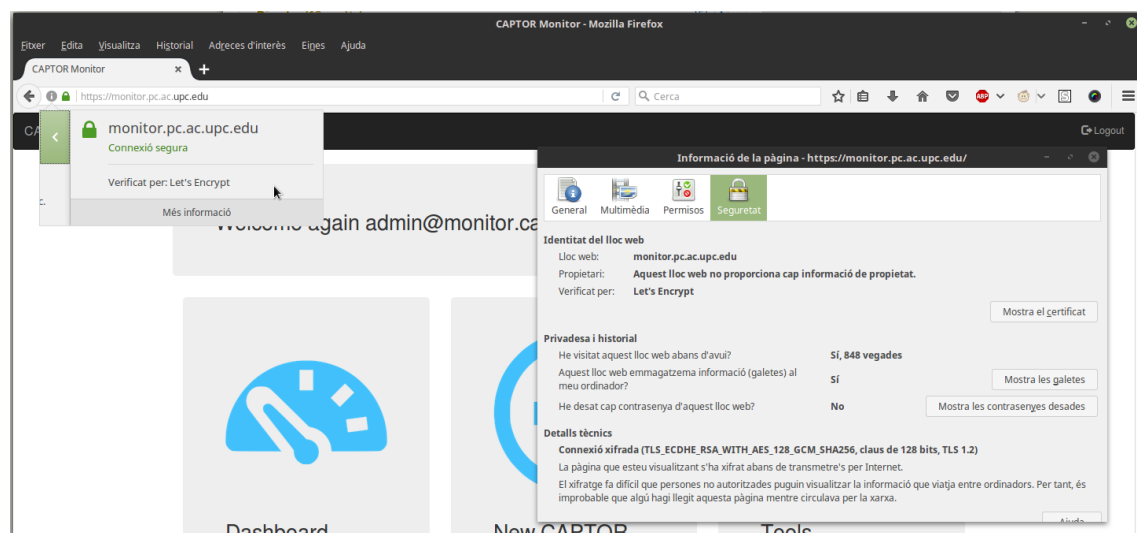


Figura 6.29: El candau verd ens informa de que es tracta d'una pàgina segura

## 6.5 Anàlisi de resultats

En aquest capítol hem construït la pàgina amb les característiques desitjades, hem configurat el servidor perquè la serveixi de la manera més eficient i l'hem assegurat amb HTTPS.

Podem auditar la seguretat de la nostra pàgina, amb el que podem comprovar que aquesta configuració es ofereix la màxima seguretat disponible.

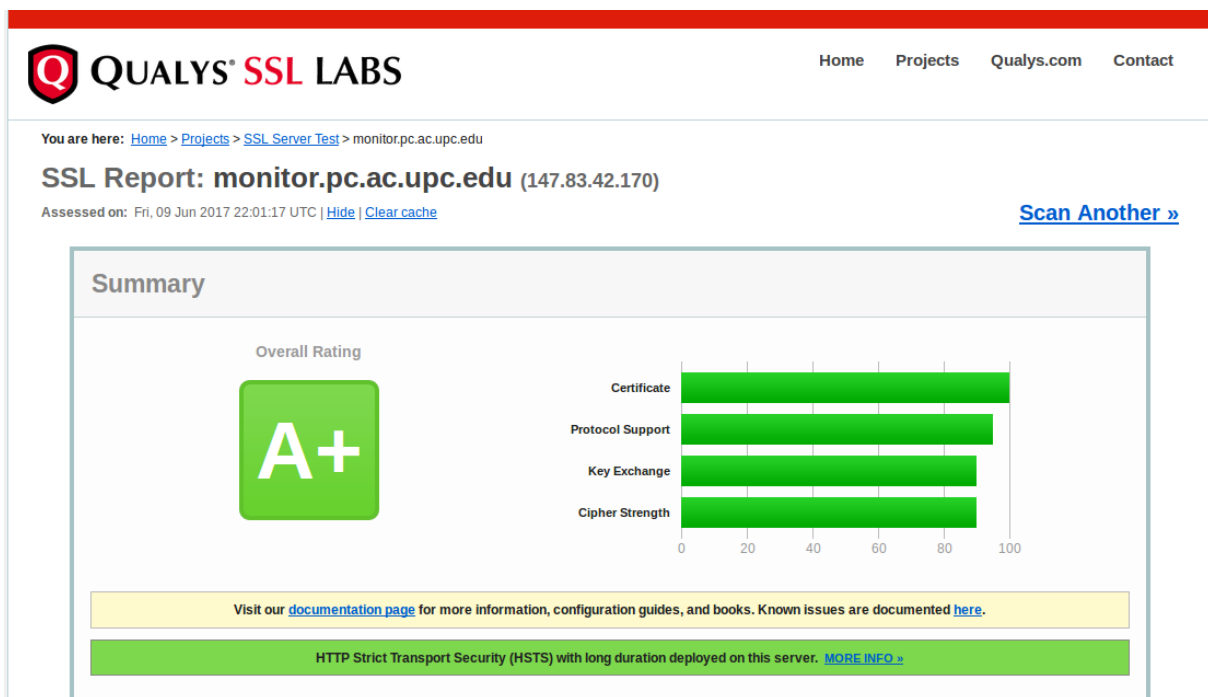


Figura 6.30: Els certificats usats ens ofereixen una seguretat molt alta

La pàgina podrà ser usada pels administradors del sistema per el propòsit plantejat, a mes, l'estructuració de la pàgina permet afegir noves funcionalitats en un futur de forma molt senzilla, pel que si en un futur es vol ampliar, serà senzill i no caldrà modificar en extrem l'estructura actual del sistema.





# Capítol 7

## Accés remot a nodes fora de la xarxa publica

### 7.1 introducció

Com hem vist en el capítol anterior, en la pàgina de monitorització, hem implementat una funció per iniciar una connexió a través d'SSH amb els nodes actius al sistema. Si els nodes disposessin d'una direcció publica, o es trobessin tots dins de la mateixa xarxa, això seria directe. A la pràctica, cada node es troba en una xarxa diferent, i es mou amb freqüenta d'emplaçament.

Ja sigui en la xarxa privada d'un voluntari, on no sempre es pot modificar la configuració de la xarxa i sols comportaria molèsties i perills de seguretat pels voluntaris, o connectat a través de la xarxa cel·lular, on directament no hi tenim control, la connexió amb els nodes no és tan directe. Per tant, per els motius que hem dit i per la comoditat de poder moure els nodes a conveniència sense haver-se de preocupar de la xarxa on estaran funcionant, hem decidit implementar una solució universal per tenir accés a aquests nodes.

El diagrama de la figura 7.1 mostra els escenaris típics de funcionament dels nodes CAPTOR. Utilitzarem la xarxa de voluntari per explicar la implementació, però el funcionament es independent de la localització del node.

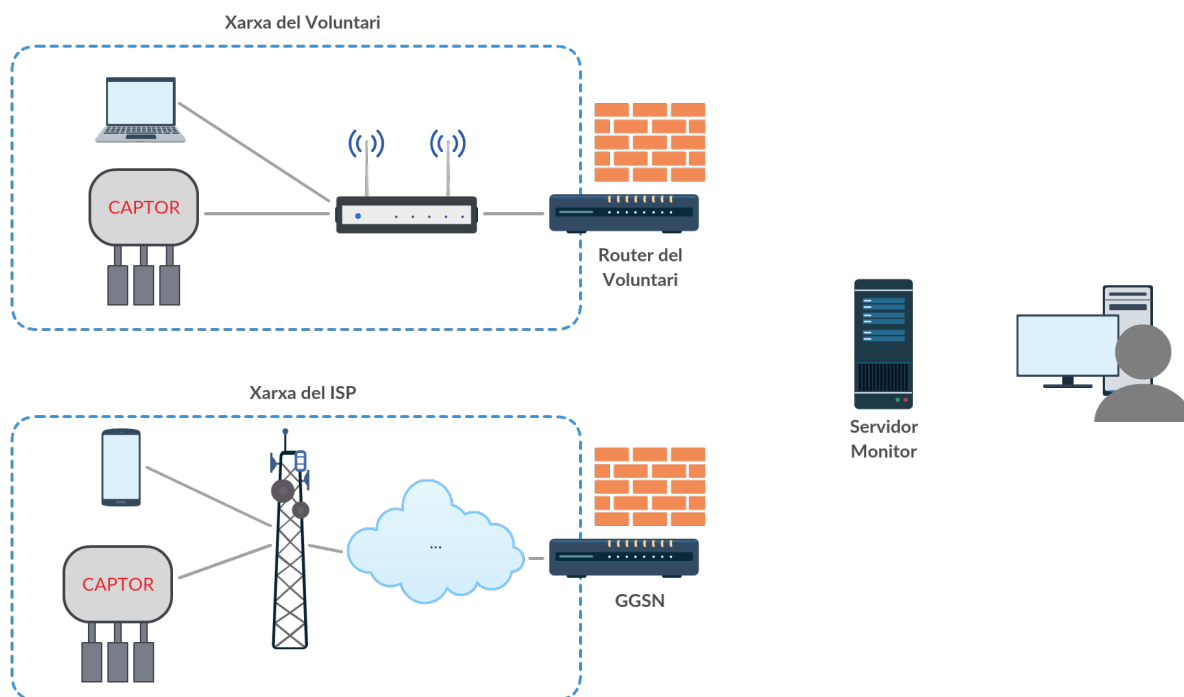


Figura 7.1: Escenari de funcionament dels nodes

## 7.2 Objectius

Els objectius d'aquesta part del treball de final de grau es poden resumir en:

- Dissenyar un sistema per accedir als nodes independentment del seu mètode d'accés a la xarxa, i que aquest mètode:
  - Sigui el màxim de passiu, limitant al necessari el tràfic en la xarxa.
  - Sigui ràpid.
  - Sigui accessible des de qualsevol dispositiu sense necessitat de configuració i instal·lació addicional (A través de l'interfície web).
  - Sigui segur.

## 7.3 Desenvolupament

### 7.3.1 Reverse SSH Tunneling

La tècnica que utilitzarem per accedir als nodes, que recordarem es troben darrere un Firewall al qual no tenim control, s'anomena Reverse SSH Tunneling o Remote Port

Forwarding. En resum, el que ens permet és establir una nova connexió SSH a través d'una connexió SSH Existent

A la que iniciem una connexió SSH a una màquina, aquesta connexió s'estableix des del client, al servidor. Un cop establerta aquesta connexió, però, podem establir una segona connexió de qualsevol tipus dins d'aquesta, amb els mateixos endpoints, però sobre la que podem decidir la direcció de la connexió, és a dir, qui dels dos actors de la connexió actua com a client i qui com a servidor. A més, podem decidir sobre quins ports actuen aquesta segona connexió, qualsevol tipus de tràfic que surti pel port indicat en el node indicat com a client, es transmetrà a través de la connexió SSH inicial al port indicat del node indicat com a servidor.

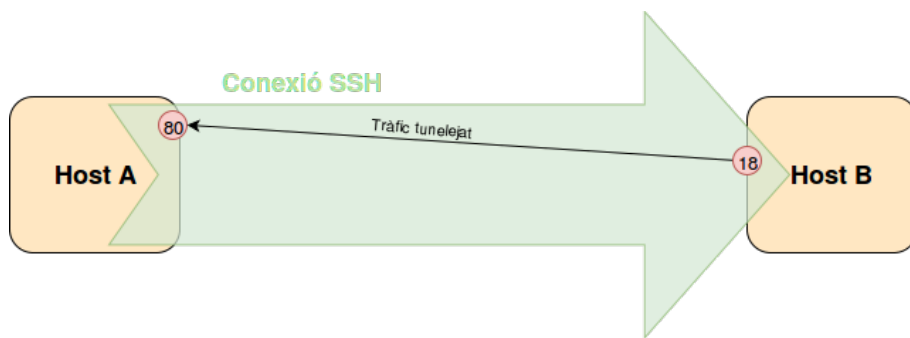


Figura 7.2: Exemple de tràfic tunelejat

Per exemple, en el diagrama de la figura 7.2, tot el tràfic que surti pel port 18 en el Host B, passarà a través del túnel fins al port 80 del Host A.

El fet més important però, és que en vistes de qualsevol observador intermediari, aquesta connexió ha estat establerta des del Host A al Host B. Per tant, si el Host A es trobés darrere un Firewall, el Firewall permetria el tràfic en la connexió ja que ha estat iniciada pel Host A.

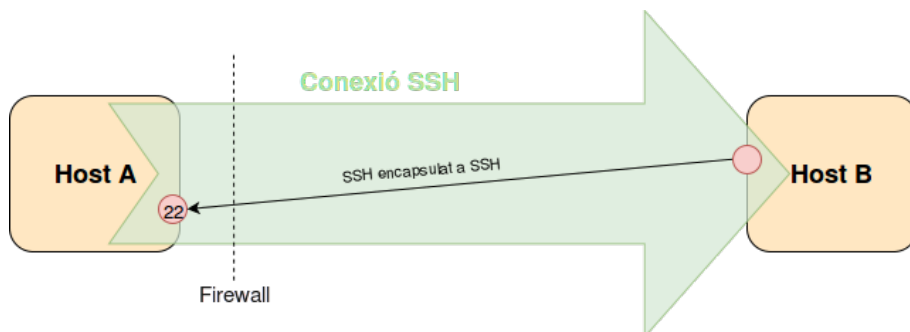


Figura 7.3: SSH Tunelejat per SSH a través del Firewall

Si el tràfic el passem cap a un port on hi hagi un servei escoltant, aquest servei tindrà un funcionament totalment corrent ja que la connexió SSH en sentit contrari es totalment Transparent. El cas que més ens interessa però, és el representat a la figura 7.3. Podem Iniciar una connexió SSH a través d'SSH superant així un firewall intermedi.

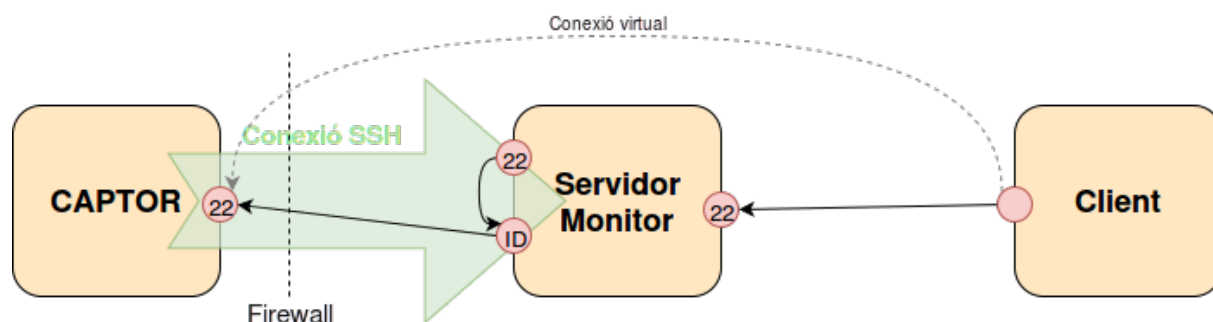


Figura 7.4: Diagrama de les connexions en el nostre sistema

Si apliquem aquest concepte al nostre escenari, podem crear una connexió virtual entre el client, accedint des de qualsevol client SSH, al CAPTOR, de forma independent a com accedeixi a la xarxa, gràcies al servidor intermediari, com podem veure a la figura 7.4.

La connexió del node al Servidor a Mes, quan no s'utilitza es totalment passiva, pel que no augmenta el tràfic de la xarxa del node si no hi ha trafic tunelejat. A mes com hem vist en el capítol anterior, podem des del servidor comprovar si hi ha una connexió en el port corresponent a la ID del node per saber si es troba funcionant correctament. Finalment, al estar tot el sistema basat en connexions SSH, disposa de la seguretat que SSH en si mateix ofereix.

### 7.3.2 Configuració d'SSH

A la figura 7.5 podem veure un resum de les connexions que establirem.

Dues d'aquestes connexions son connexions regulars per SSH:

- **Connexió del client al servidor:** Per iniciar la connexió amb el node, hem de primer connectar-nos al servidor intermediari Monitor. Aquest pas no és necessari si accedim des de l'interfície web, ja que el servidor s'encarrega de fer-ho, però el nostre sistema es capaç d'iniciar una sessió amb el node a través d'una terminal estàndard. Des d'una consola al client: `ssh usuari@monitor.pc.upc.edu`.
- **Connexió dins del servidor:** Aquesta connexió es realitza dins del mateix servidor i serveix per comunicar la connexió del client amb el túnel Reverse SSH, a la pràctica, estem creant una connexió que comunica la sortida de la connexió client a la entrada de la connexió que passa a través del túnel. Des de la connexió anterior al servidor

Monitor: `ssh root@localhost -p [ID node]`. Usem l'usuari root ja que és l'únic usuari en els nodes, i amb el paràmetre -p, connectem la connexió en el port, on, com ja veurem, hi haurà un túnel escoltant.

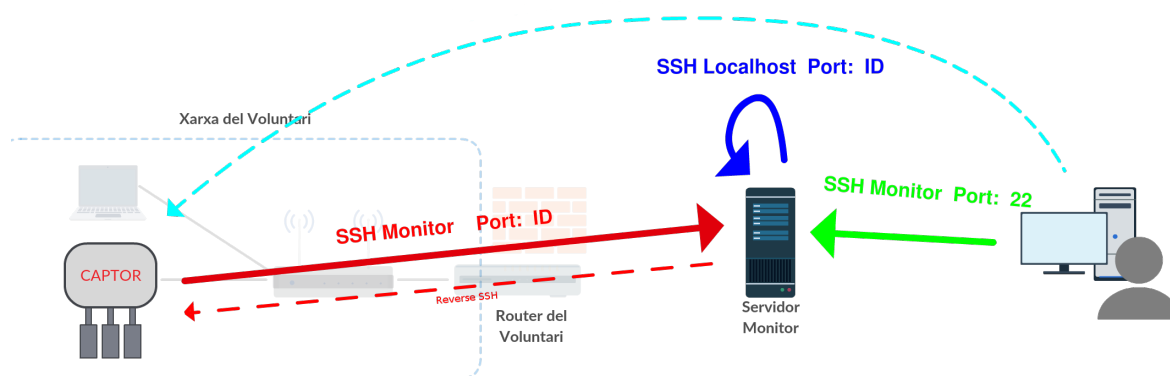


Figura 7.5: Resum de les connexions

La tercera connexió es realitzarà des del node CAPTOR cap al servidor de monitor cada vegada que es trobi online. A més aquesta connexió servirà de túnel a la connexió SSH en sentit contrari. SSH ofereix uns paràmetres que permeten automàticament, iniciar una connexió i crear un canal dins d'aquesta connexió a l'espera de rebre missatges. Per tant, la comanda que executarem al node CAPTOR serà la següent:

```
ssh -R [ID node]:localhost:22 captor@monitor.pc.ac.upc.edu
```

- **-R** : Amb aquest paràmetre indiquem que volem que crei un canal dins de la nostra connexió SSH amb l'extrem on s'escolta al Remot, és a dir, que crei un canal amb el servidor de Monitor com a client i el CAPTOR com a servidor. Amb el paràmetre -L indicariem que volem que l'extrem a escoltar sigui el Local, un túnel en la direcció contrària.
- **[ID node]:localhost:22** : Indiquem que volem que el túnel connecti el port [ID node] del remot amb el port 22 (SSH) local.

Per tant, establint la connexió des del node al Servidor i des del client al servidor, podem iniciar una connexió entre el client i el node a través del túnel que hem creat.

### 7.3.3 Automatització de la connexió del CAPTOR

Ja que desitjem que aquesta connexió entre el node i el Servidor sigui permanent, ens hem d'assegurar de que:

- La connexió es reinicii si SSH falla.
- La connexió es reinicii si la xarxa falla i es recupera.
- La connexió s'iniciï automàticament al iniciar el node.

Els dos primers punts, com veurem, els solucionarem a la vegada monitoritzant la connexió i re-iniciant-la quan calgui, pel que fa al tercer, haurem de crear un servei Linux per al node que s'encarregui d controlar aquest canal.

### 7.3.3.1 Monitoritzant la connexió amb AutoSSH

AutoSSH es un programa que s'inclou amb OpenWRT que s'encarrega de monitoritzar connexions SSH i reiniciar-les si la connexió falla.

Com hem dit, la comanda que utilitzarem per monitoritzar la connexió és la següent:

```
ssh -R [ID node]:localhost:22 captor@monitor.pc.ac.upc.edu
```

Per monitoritzar-la amb AutoSSH, es tant simple com afegir la comanda al davant:

```
autossh -R [ID node]:localhost:22 captor@monitor.pc.ac.upc.edu
```

Tot i així, hem de tenir en compte com funciona AutoSSH, ja que el seu comportament no és exactament el desitjat. Amb el comportament per defecte, AutoSSH utilitzarà dos ports de monitorització (l'indicat per el paràmetre -M i el següent, per defecte els dos consecutius al port de la connexió monitoritzada) per enviar missatges contínuament entre el client i el servidor de la connexió. Enviarà dades per el primer i rebrà respostes pel següent.

Aquest comportament, en el nostre escenari no és del tot desitjable, volem que la connexió sigui el màxim de passiva i a més, ens hem d'assegurar que els ports que utilitzarà AutoSSH per monitorització estiguin lliures tant en el captor com al servidor. Ja que no podem estar segurs quin nombre de connexions en ports consecutius tindrà el servei Monitor (de cada un dels nodes de la xarxa), és una opció poc pràctica. Així que desactivarem aquesta característica amb la comanda -M 0.

```
autossh -M 0 -R [ID node]:localhost:22 captor@monitor.pc.ac.upc.edu
```

### 7.3.3.2 Daemon per iniciar la connexió

Com hem dit, desitgem que la connexió, a més s'iniciï automàticament al iniciar el node. Els nodes, degut a que el seu sistema operatiu esta basat en openWRT, no utilitzen systemd com a sistema de boot, si no procd, un sistema de boot específic. Per tant, no podem usar un servei de systemd com hem fet per servir la pàgina web. Així que crearem un daemon que serà iniciat per init.d (a través de Busybox) en temps de boot.

---

```
1  #!/bin/sh /etc/rc.common
2  # Monitor init script
3  # CAPTOR AC UPC 2017
4  # @author: aina
5
6  START=99
7  STOP=1
8
9  CAPTOR_ID=17021
10 RSA_KEYS_PATH=/root/.ssh/id_rsa
11 USERNAME=captor
12 SERVER_MONITOR=monitor.pc.ac.upc.edu
13
14 start() {
15
16     echo "starting monitor service"
17     autossh -M 0 -f -NT -i $RSA_KEYS_PATH -R $CAPTOR_ID:localhost:22
18     ↪ $USERNAME@$SERVER_MONITOR&
19 }
20
21 stop() {
22     echo "stopping monitor service"
23     killall autossh
24 }
```

---

#### Codi 35: Primer init Script de Monitor

El codi 35 es troba en el fitxer */etc/init.d/monitor*.

Com podem veure, utilitza */etc/rc.common* com a wrapper. El fet de que utilitzem aquest wrapper, s'assegurarà de que el script no tingui errors que puguin fer que el sistema s'iniciï incorrectament, així com que disposi de com a mínim de les funcions d'start i d'stop. Podem activar aquest daemon gràcies a aquest wrapper, que a més ens proporciona aquesta funcionalitat, de la següent manera:

---

```

1  #Iniciar el daemon
2  /etc/init.d/monitor start
3
4  #Aturar el daemon
5  /etc/init.d/monitor stop
6
7  #Cridar consecutivament start i stop
8  /etc/init.d/monitor restart

```

---

Codi 36: Activant, desactivant i reiniciant el daemon de Monitor

El procés init d'openWRT, com a pas final, executarà tots els fitxers que trobi a `/etc/rc.d/` amb la opció "start". A `rc.d` hi haurà links simbòlics als scripts, numerats segons el valor de la variable `START` que hem definit en el seu interior. Aquestes variables, `START` i `STOP`, ens indiquen l'ordre en el que s'executaran i pararan tots els daemons. Ja que volem que el nostre s'executi cap al final quan la resta de serveis estiguin ja iniciats, i es pari dels primers, abans que la resta de serveis, posem el màxim valor a `START` i el mínim a `STOP`.

A continuació, usem la comanda `/etc/init.d/monitor enable` per crear aquest link simbòlic. A la figura 7.6 podem veure que s'ha afegit a `/etc/rc.d` un enllaç al nostre script anomenat `S99monitor`.

```

root@captor17021:~# ls /etc/rc.d/
K15captor      S48rngd
K1monitor      S49delete_uhttpd_cert
K50dropbear    S49generate_new_gpg_key
K90network     S50cron
K95luci_fixtime S50dropbear
K98boot        S50uhttpd
K99umount      S59luci_dhcp_migrate
K99usd         S60dbus
S05defconfig   S60dnsmasq
S05luci_fixtime S61avahi-daemon
S09handle_wifi_reset S80autossh
S10boot        S93triggerhappy
S10captor      S95done
S11sysctl      S96led
S11ubus        S97watchdog
S18rename-wifi-if-access-point S98sysnptd
S19firewall    S99monitor
S20fstab       S99rngd-turn-off
S20network     S99usd
S39usb

```

Figura 7.6: Continguts de `/etc/rc.d` del node

Amb això, aconseguim que al iniciar-se el node, s'iniciï aquest daemon, que controlarà la comunicació amb el sistema.



### 7.3.3.3 Claus RSA i Known Hosts

Ja que la comunicació s'iniciarà automàticament i no es podran entrar manualment credencials i confirmacions, encara falta una última cosa perquè el nostre daemon funcioni correctament. Afegir les claus RSA perquè el node es pugui connectar al servidor de Monitor sense contrasenya i la firma del servidor a la llista de hosts coneguts perquè no demani confirmació.

Pel que fa a les claus, hem de modificar el fitxer `.ssh/authorized_keys` perquè contingui les noves claus:

---

```
1 ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQg[...]EXF/6nUe78fME4sjGDKoPN9q9Siah8nuxz2X1q79Ztnk
2 = root@captor17021
```

---

Codi 37: continguts de `.ssh/authorized_keys`

Per a la firma del servidor, hem de modificar el fitxer `known_hosts`:

---

```
1 monitor.pc.ac.upc.edu ssh-rsa AAAAB3N[...]BpzI8h9
```

---

Codi 38: continguts de `known_hosts`

### 7.3.3.4 Problemes amb la connexió

Tot i amb això, ens trobàvem que la connexió es tancava al costat del servidor de Monitor sense aparent motiu ni explicació mentre seguia funcionant de forma esperada al cantó del node CAPTOR (Per tant, no era el servidor que tancava la connexió, ja que AutoSSH ho hauria detectat i reiniciat). Com ja hem dit, el servidor es troba en una màquina virtual del departament d'Arquitectura de Computadors de la UPC, així que la única explicació que hem trobat és que el firewall del departament oblida la connexió al cap d'unes hores per trobar-se inactiva. Per solucionar això, hem de modificar la comanda d'AutoSSH, de forma que l'script del daemon definitiu queda així:

---

```
1  #!/bin/sh /etc/rc.common
2  # Monitor init script
3  # CAPTOR AC UPC 2017
4  # @author: aina
5
6  START=99
7  STOP=1
8
9  CAPTOR_ID=17021
10 RSA_KEYS_PATH=/root/.ssh/id_rsa
11 USERNAME=captor
12 SERVER_MONITOR=monitor.pc.ac.upc.edu
13
14 start() {
15
16     echo "starting monitor service"
17     autossh -M 0 -o "ServerAliveInterval 60" -f -NT -i $RSA_KEYS_PATH -R
18     ↪ $CAPTOR_ID:localhost:22 $USERNAME@$SERVER_MONITOR&
19 }
20
21 stop() {
22     echo "stopping monitor service"
23     killall autossh
24 }
```

---

#### Codi 39: Daemon de Monitor

Com podem veure, hem afegit el paràmetre *-o "ServerAliveInterval 60"*. Aquest paràmetre fa que AutoSSH envii cada 60 minuts un paquet buit a través de la connexió, cosa que fa que el firewall la consideri com a activa i no la tanqui, a més, ajuda a que la connexió sigui més estable, ja que si AutoSSH no rep resposta d'aquest paquet, reinicia la connexió.

Amb això aconseguim que la connexió amb el servidor de monitor sigui totalment estable.

## 7.4 Alternativa: SIM Emnify

Durant l'implementació d'aquest sistema, es decideix canviar la companyia que proveeix la connexió 3G als nodes. Es decideix contractar una empresa (Emnify) orientada a oferir

connexió a dispositius intel·ligents a diferència de les companyies tradicionals orientades a oferir connexió al consumidor final. Aquesta companyia ofereix a més un servei extra: Totes les SIM contractades pel mateix client es trobaran dins d'una mateixa VPN a la qual el client té accés.

Per tant, a través de la VPN aquesta, som capaços d'accedir als nodes com si es trobessin dins de la nostra pròpia xarxa local.

Aquesta alternativa ens suposa el benefici de que no hem d'instal·lar res addicional als nodes, però creiem que la nostra solució segueix sent millor pels següents motius:

- La connexió és la mateixa pels nodes que funcionen amb 3G i els que funcionen amb WiFi sense distinció
- La connexió és significantment més fluida ja que no tenim el overhead de la VPN
- No és necessari instal·lar una VPN al nostre dispositiu, per tant es pot usar des de qualsevol dispositiu i sistema operatiu que disposi d'un client SSH o a través de la pàgina web.
- La instal·lació és igualment automàtica (s'inclou als scripts d'instal·lació dels nodes) i no suposa feina extra

## 7.5 Anàlisi de resultats

En aquest capítol, hem creat una connexió permanent entre el node i el servidor del servei Monitor que ens proporciona les següents característiques:

- Ens permet comunicar-nos entre el Servidor i el node.
- Ens permet crear una connexió virtual entre el client i el node a través del servidor.
- Ens permet monitoritzar l'estat de la connexió amb el node (i per tant l'estat del node).
- És més eficient i ràpida que les seves alternatives.

Per tant, creiem que hem assolit els objectius esperats per a aquest apartat del Treball de Final de Grau, i que aquesta connexió servirà als administradors dels nodes per administrar a distància i em comoditat els diferents CAPTOR de la xarxa.



# Capítol 8

## Conclusions

### 8.1 Conclusions

Com ja hem vist amb més detall als corresponents apartats dels capítols anteriors, creiem que el resultat del treball ha sigut molt positiu i hem assolit els nostres objectius.

La placa que hem realitzat per millorar la presa de dades dels nodes funciona perfectament i de la forma esperada, i a més hem aconseguit reduir costos de fabricació del node. A més, aporta un aspecte més professional i millora l'estabilitat dels components al ser transportats, pel que facilita molt el manteniment per part dels voluntaris un cop rebel el node al seu país.

Creiem que la pàgina web que hem realitzat, assoleix tots els objectius proposats en un principi, de forma eficient, intuïtiva i agradable, i creiem que hem aconseguit una eina útil que facilitarà el manteniment per part dels responsables i el coneixement per part dels voluntaris.

Finalment creiem que el sistema de connexió que hem dissenyat és la millor opció possible, hem aconseguit que sigui el màxim d'estable amb els mínims recursos, i que el seu funcionament sigui senzill, transparent, potent, i flexible. També creiem, pels motius que ja hem detallat en el apartat corresponent, a les conclusions del capítol pertanyent, que la nostra solució és per diversos motius més útil que l'alternativa proposada per l'ISP.

#### 8.1.1 Relació amb el grau

Hem intentat, amb aquest treball, cobrir gran part de les competències vistes al grau:

- Hem tractat temes de hardware, vistos en assignatures com Interfícies de Computadors (CI).

- Hem tractat també temes de disseny de serveis web, vistos en assignatures com Aplicacions Distribuïdes (AD) i Projecte de Tecnologia de l'Informació (PTI).
- Hem dissenyat i utilitzat bases de dades gracies als coneixements obtinguts a assignatures com Bases de Dades (BD) i a Enginieria del Software (IES).
- Hem utilitzat coneixements sobre xarxes obtinguts a les assignatures de Xarxes de Computadors (XC), Protocols d'Internet (PI), Internet Mòbil (IM) i Tecnologies de Xarxes de Computadors (TXC).
- Hem pogut fer el sistema segur tenint en compte qüestions vistes a Seguretat Informàtica (SI)
- I hem utilitzat els coneixements sobre Linux i sistemes operatius en general obtinguts en assignatures com Sistemes Operatius (SO), Administració de Sistemes Operatius (ASO), Sistemes Operatius Avançats (SOA) i Conceptes Avançats de Sistemes Operatius (CASO).
- Finalment, hem fet us de competències genèriques aconseguides gràcies a les assignatures de programació i software lliure.

### 8.1.2 Aspectes a millorar

Tot i el èxit que creiem haver assolit en la major part dels objectius del treball, creiem que hi ha alguns punts que potser no han acabat de quedar del tot tancats i s'haguessin pogut fer d'una altre manera o necessiten treball per funcionar òptimament:

- La pàgina web en producció no és del tot estable si la comparem a la pàgina web en desenvolupament, això pot ser degut a la configuració del servidor d'aplicacions, que pot ser que no sigui la òptima.
- La funció de connexió per SSH a través del navegador es troba separada de la pàgina web. No és específicament una inconveniència, però potser hagués sigut interessant integrar-ho tot en el mateix component per millorar l'estabilitat de tot el conjunt.
- El Daemon que controla la connexió, envia un petit paquet buit per hora cap al servidor, això ha sigut degut a un arranjament d'última hora per sobrepassar un inconvenient inesperat, com ja hem explicat. Si el servei hagués estat allotjat en una xarxa en la que tinguéssim control absolut, segurament haguéssim pogut modificar la configuració per evitar la necessitat d'aquests paquets.

## 8.2 Opinió Personal

Personalment, estic molt contenta amb el resultat d'aquest treball. M'ha permès aprofundir en temes que m'interessàvem però que potser no havia acabat de veure en detall durant el grau, a més, m'ha permès conèixer tot un seguit d'eines i projectes molt interessants que probablement em seran útils durant la meva carrera professional.

Treballar en un projecte gran, amb projectes de codi lliure i ciència ciutadana, que treballa per millorar el medi ambient i la qualitat de vida dels habitants del planeta és a més una experiència molt satisfactòria, i m'ha permet conèixer a gent de molts àmbits dels que he pogut aprendre sobre camps que no m'hagués plantejat mai.

En definitiva, crec que aquest treball ha sigut molt positiu tant com per recopilar i posar en pràctica coneixements assolits durant el grau, com per donar-me a comprendre millor noves eines i conceptes molt actuals i útils. Una experiència productiva de la que ha sorgit un treball el resultat del qual n'estic orgullosa.

## 8.3 Propostes de Terball Futur

El projecte en sí ha quedat molt tancat, però la proposta inicial era molt ambiciosa i va quedar clar que havíem de retallar certes funcions que desgraciadament no podrien implementar-se a la versió presentada. També durant el transcurs d'implementació, van sorgir noves idees que no totes van poder ser afegides.

Creiem que l'aspecte del treball que te mes marge per ser ampliat i créixer es la pàgina web, especialment la part dedicada al public general mes que als administradors. A continuació, detallo algunes proposta i notes sobre la proposta de característiques que es podrien afegir:

- A la fitxa de cada node, es pot afegir la gràfica de dades preses durant els últims dies. Aquestes dades i gràfiques ja es poden veure a través de consensum, però no existeix una API que permeti consultar-les. Una idea seria obtenir-les a partir d'analitzar el HTML de la pàgina que et mostra les dades enlloc de de la API.
- A la fitxa de cada node, també, es podria afegir un llistat de notes o apunts per als administradors del sistema, com petits blocs de text que es poden afegir o eliminar en qualsevol moment per tenir una eina on apuntar modificacions, notes i avisos de forma auxiliar. La base de dades ja està preparada per a aquest propòsit.
- Podria ser interessant afegir un mapa d'on es troba cada node, on es vaguessin tots (ja es disposa de les dades de posició de cada node) potser amb diferents modes per veure dades dels sensors també representades sobre el mapa.





# Bibliografia

- [1] OMS (Organización Mundial de la Salud). *Calidad del aire ambiente (exterior) y salud*. 2016. URL: <http://www.who.int/mediacentre/factsheets/fs313/es/> (cons. 25-02-2017).
- [2] OMS (Organización Mundial de la Salud). *Learn more about pollution*. 2016. URL: <https://ncar.ucar.edu/learn-more-about/pollution> (cons. 25-02-2017).
- [3] Departament de Territori i Sostenibilitat. *Què és l'ozó troposfèric?* 2014. URL: [http://mediambient.gencat.cat/ca/05\\_ambits\\_dactuacio/atmosfera/qualitat\\_de\\_laire/avaluacio/campanya\\_ozo/que\\_es\\_lozo\\_troposferic/](http://mediambient.gencat.cat/ca/05_ambits_dactuacio/atmosfera/qualitat_de_laire/avaluacio/campanya_ozo/que_es_lozo_troposferic/) (cons. 25-02-2017).
- [4] Pedro Belmonte i Eduardo Gutiérrez. *Ozono troposférico*. 2013. URL: <http://www.ecologistasenaccion.org/article27108.html> (cons. 25-02-2017).
- [5] Projecte CAPTOR. *Collective awareness platform for tropospheric ozone pollution*. 2016. URL: <https://www.captor-project.eu/index.php/en/>.
- [6] PandoraFMS. *Monitorización de Sistemas Informáticos, ventajas, procedimientos e implementación*. 2016. URL: <https://blog.pandorafms.org/es/monitorizacion-de-sistemas/> (cons. 25-02-2017).
- [7] Emnify. *IoT & M2M connectivity management platform*. 2017. URL: <https://www.emnify.com/> (cons. 28-05-2017).
- [8] W3c. *Server-Sent Events Especification*. 2015. URL: <https://www.w3.org/TR/eventsource/> (cons. 28-05-2017).
- [9] Krishna Srinivas. *Terminal in browser over http/https*. 2016. URL: <https://github.com/krishnasrinivas/wetty> (cons. 28-05-2017).
- [10] Justin Ellingwood - Digital Ocean Deployment team. *How To Serve Flask Applications with uWSGI and Nginx*. 2016. URL: <https://www.digitalocean.com/community/tutorials/how-to-serve-flask-applications-with-uwsgi-and-nginx-on-ubuntu-16-04> (cons. 28-05-2017).
- [11] Phillip J. Eby. *Python Web Server Gateway Interface v1.0*. 2003. URL: <https://www.python.org/dev/peps/pep-0333/> (cons. 28-05-2017).



# Índex de figures

2.1	Estació de referencia . . . . .	5
2.2	Formació de l'ozó troposfèric . . . . .	6
2.3	Zones amb l'aire mes contaminat(2011) . . . . .	7
2.4	Sensor CAPTOR . . . . .	8
3.1	Sistema PandoraFMS de monitoratge . . . . .	11
5.1	Calibració CAPTOR 16002 . . . . .	28
5.2	Calibració CAPTOR 16002 . . . . .	29
5.3	Calibració CAPTOR 16005 . . . . .	30
5.4	Calibració CAPTOR 16005 . . . . .	30
5.5	Arduino Yun . . . . .	32
5.6	Bridge entre els dos entorns de l'Arduino Yun . . . . .	32
5.7	Logotip d'OpenWRT . . . . .	33
5.8	Logotip d'EAGLE . . . . .	33
5.9	Proces d'aplicació de la màscara de tòner . . . . .	35
5.10	Connexions internes del CAPTOR durant la campanya del 2016 . . . . .	36
5.11	Diagrama de connexions del que es parteix per realitzar l'esquema . . . . .	37
5.12	Diagrama auxiliar de les connexions . . . . .	37
5.13	Diagrama final del prototip . . . . .	38
5.14	Placa finalitzada i serigrafia . . . . .	39
5.15	Prototips finals . . . . .	39
5.16	Diagrama final de la placa del CAPTOR . . . . .	40
5.17	Placa final per als CAPTOR de la campanya 2017 . . . . .	41

5.18	Procés de muntatge dels nodes CAPTOR . . . . .	41
5.19	Segona cal·libració del CAPTOR 17013 . . . . .	42
5.20	Node final . . . . .	43
6.1	Logotip Python . . . . .	46
6.2	Logotip Flask . . . . .	47
6.3	Logotip NodeJS . . . . .	50
6.4	Logotip NGINX . . . . .	51
6.5	Logotip uWSGI . . . . .	51
6.6	Logotip Let's Encrypt . . . . .	52
6.7	Logotip PostgreSQL . . . . .	52
6.8	Logotip SQLAlchemy . . . . .	53
6.9	Estructura de la pàgina web, amb la URL i els permisos necessaris . . . . .	54
6.10	Estructura de la base de dades . . . . .	55
6.11	Estructura del directori monitor . . . . .	59
6.12	Contingut del directori templates . . . . .	60
6.13	Vista de la pàgina d'informació per un usuari regular . . . . .	67
6.14	Vista de la pàgina d'informació per un usuari administrador . . . . .	67
6.15	Vista del formulari per afegir un node . . . . .	68
6.16	API de Google . . . . .	71
6.17	Vista del formulari per editar un node . . . . .	73
6.18	Avis al eliminar un node . . . . .	74
6.19	Dashboard amb nodes connectats del tot, amb problemes de 3G i desconnectats	75
6.20	Estat dels ports al servidor . . . . .	75
6.21	Contingut de <i>/proc/net/tcp6</i> . . . . .	76
6.22	Missatges enviats en un SSE . . . . .	77
6.23	Missatges rebuts cada 2 segons . . . . .	78
6.24	Sequencia d'autorització d'aplicació a Emnify . . . . .	79
6.25	Solicitud de Token d'aplicació . . . . .	80
6.26	Sessió SSH establerta amb el node . . . . .	82
6.27	Funcionament d'un certificat de Let's Encrypt en el servidor Nginx . . . . .	85

---

6.28	Generant un grup Diffie-Hellman de 2048 bits . . . . .	86
6.29	El candau verd ens informa de que es tracta d'una pàgina segura . . . . .	88
6.30	Els certificats usats ens ofereixen una seguretat molt alta . . . . .	89
7.1	Escenari de funcionament dels nodes . . . . .	92
7.2	Exemple de tràfic tunelejat . . . . .	93
7.3	SSH Tunelejat per SSH a través del Firewall . . . . .	93
7.4	Diagrama de les connexions en el nostre sistema . . . . .	94
7.5	Resum de les connexions . . . . .	95
7.6	Continguts de /etc/rc.d del node . . . . .	98



# Índex de codi

1	Estructura bàsica d'una aplicació web en Flask . . . . .	47
2	Exemple de Jinja2 . . . . .	48
3	Exemple d'SSE, Client side . . . . .	49
4	Exemple d'SSE, Server side . . . . .	50
5	Definim la connexió amb la base de dades . . . . .	56
6	Creem les classes necessàries per a la base de dades . . . . .	58
7	Afegim elements i consultam la base de dades . . . . .	58
8	run_external.py . . . . .	61
9	Exemple de servir una pàgina en Flask . . . . .	61
10	Fragment HTML de vista de CAPTOR . . . . .	64
11	Exemple de If/Else . . . . .	64
12	Servint la pàgina d'afegir CAPTORs amb restricció a administradors . . .	65
13	Informat a Jinja2 que afegeixi el contingut d'administrador . . . . .	65
14	Fragment d'HTML que només s'inclou si s'és administrador . . . . .	66
15	Classes CSS del desplegable . . . . .	69
16	Recollim per post i inserim a la base de dades. Fragment . . . . .	70
17	Identificant-nos a la API de Google Maps . . . . .	71
18	Creant un Mapa, un marcador i utilitats del mapa . . . . .	72
19	Eliminant un CAPTOR de la base de dades . . . . .	74
20	Obtenint nodes connectats . . . . .	76
21	Petició de subscripció al event . . . . .	77
22	Resposta a la petició d'event stream . . . . .	78
23	Server-Sent Event stream . . . . .	78
24	Obtenció d'un token d'autorització . . . . .	80
25	Obtenint informació desitjada de la API . . . . .	80
26	Iniciant una sessió SSH . . . . .	81
27	WSGI Entry Point Descriptor . . . . .	82
28	WSGI Config File . . . . .	83
29	Contingut de monitor.service . . . . .	84
30	Activant el servei . . . . .	84
31	Configuració del block de servidor NGINX . . . . .	85

32	Configuració del block de servidor NGINX . . . . .	86
33	Iniciant el servei de certbot . . . . .	86
34	Configuració del block de servidor NGINX, amb TLS/SSL . . . . .	87
35	Primer init Script de Monitor . . . . .	97
36	Activant, desactivant i reiniciant el daemon de Monitor . . . . .	98
37	continguts de .ssh/authorized_keys . . . . .	99
38	continguts de known_hosts . . . . .	99
39	Daemon de Monitor . . . . .	100